



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

1965-11-01

A FORTRAN program for permuted title indexing.

Drury, William R.

University of Kansas

---

<http://hdl.handle.net/10945/12237>

---

*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>

NPS ARCHIVE  
1965  
DRURY, W.

# A FORTRAN PROGRAM FOR PERMUTED TITLE INDEXING

WILLIAM R. DRURY

1965

Library  
U. S. Naval Postgraduate School  
Monterey, California

DUDLEY KNOX LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY CA 93943-5101

1500/Na13  
Sec: 577.3

U. S. NAVAL RESERVE OFFICERS TRAINING CORPS  
THE UNIVERSITY OF KANSAS  
LAWRENCE, KANSAS

IN REPLY REFER TO:  
NROTCU/HHD:rr  
1520  
7 December 1965

From: Commanding Officer, NROTC Unit, University of Kansas, Lawrence, Kansas  
To: Superintendent, Postgraduate School, Monterey, California

Subj: Completion of course report; forwarding of

Ref: (a) USNPGS INST P5000.2B of 14 FEB 65

Encl: (1) Thesis (2 copies) - 1 Copy only for 038.  
(2) Official Transcript  
(3) NPS Form #100  
(4) Copy of Bupers Order 152394 w/endorsements

1. In accordance with reference (a), enclosures (1) through (4) are forwarded herewith for the below listed officer:

LT William R. DRURY, 624483/3100, USN













A FORTRAN PROGRAM FOR PERMUTED TITLE INDEXING

by

William R. Drury

//  
B.S., United States Naval Academy, 1958

Submitted to the Department  
of Chemical and Petroleum  
Engineering and the Faculty  
of the Graduate School of  
The University of Kansas in  
partial fulfillment of the  
requirements for the Degree  
of Master of Science

November 1965



## ABSTRACT

Raw data pertaining to thesis topics in the fields of Chemical and Petroleum Engineering, submitted to 144 universities and other institutions of higher learning, are indexed by keywords-in-context and displayed permutedly. The printed output reference listing is entered by keyword. Serial references to author, school, and hardcopy may be read directly.

Reference listings were prepared using an IBM 7040 Fixed-word-length Computer. The computer programs are written in Fortran (IV) computer language.

PROPERTY OF THE  
NATIONAL POSTAL SERVICE  
UNITED STATES OF AMERICA

## ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to Dr. Floyd W. Preston whose guidance, wise counsel, patience, and humor were ever-present in bringing this thesis to completion.

Also, Dean Wiley S. Mitchell, Associate Dean of the School of Business, and Dr. Charles F. Weinaug, Chairman of the Department of Petroleum Engineering, are due many thanks for their efforts on behalf of the Petroleum Management program. To Mr. Maung Maung Win of the Chemical Engineering Department are extended sincere expressions of appreciation for the kindness shown in permitting the author to use his extensive data collection for this study.

Special recognition is given to the United States Navy whose sponsorship made this study possible.



## TABLE OF CONTENTS

Chapter	Page
I INTRODUCTION.....	1
Purpose.....	1
Description of Terms.....	1
Permuted Title Indexing.....	2
Origin.....	3
II CONSIDERATIONS IN ESTABLISHING AN INFORMATION RETRIEVAL SYSTEM.....	5
General.....	5
Additional Considerations.....	6
Implications to Military Situations.....	8
III A PROBLEM IN PERMUTED TITLE INDEXING.....	10
Data.....	10
Method of Solution.....	10
Record Processing.....	11
Error Detection.....	11
Keyword Identification.....	12
Tape Loading.....	13
Sorting.....	13
Write-out Procedures.....	14
Description of Programs.....	14
IV ANALYSIS OF PROCEDURES AND COMPUTER TECHNIQUES....	20
Utilization of Computer Time.....	21
Interpretation of Output.....	24
V CONCLUSION.....	25
Summation of Findings.....	25
Projections of the Future.....	26





# TABLE OF CONTENTS (continued)

Chapter	Page
Food for Thought.....	26
REFERENCES.....	28
BIBLIOGRAPHY.....	29
APPENDIX A.....	31
Fortran Notation.....	31
Flow Chart for Mainline Program.....	33
Source Deck Listing (Mainline Program).....	39
Flow Chart for Subroutine KICK.....	45
Source Deck Listing (KICK Subroutine).....	46
Flow Chart for NBRT Subprogram.....	47
Flow Chart for KSKOL Subprogram.....	47
Source Deck Listing (NBRT Subprogram).....	48
Source Deck Listing (KSKOL Subprogram).....	49
Flow Chart for Master Tape Write-out.....	50
Source Deck Listing (Master Write-out).....	51
Flow Chart for Permuted Title Index Write-out.....	52
Source Deck Listing (Permuted Title Index Write-out).....	54
Flow Chart for Author Listing Write-out.....	56
Source Deck Listing (Author Write-out).....	57
Source Deck Listing (PACK Subroutine).....	58
Source Deck Listing (UNPACK Subroutine).....	58
Control Cards for SORT.....	59
Discard Dictionary.....	60
APPENDIX B.....	61



## TABLE OF CONTENTS (continued)

Chapter	Page
Sample Input Data.....	61
APPENDIX C.....	62
School Dictionary.....	62
Permuted Title Index.....	62
Master Record Bibliographic Listing.....	62



## LIST OF FIGURES

Figure	Page
1 Sample Record Formats.....	16
2 Generalized Flow Chart.....	19



# Chapter I

## INTRODUCTION

Purpose of Investigation Information accumulation, once the very object of man's formal education, is rapidly becoming a giant boulder to be thrown in the path of man's search for meaningful data. The volume of printed matter on virtually any subject grows hourly at a swift pace and man has been slow to grasp this reality.

This thesis deals with one aspect of information handling wherein computer processing techniques are utilized to perform cataloging, alphabetizing, indexing, and printing of a large volume of historical data pertaining to writings in a specialized field of endeavor (ie: Chemical and Petroleum Engineering). It is the author's aim to determine whether it is feasible to create a permuted title index with an engineering oriented, fixed-word-length digital computer. If so, such a program will be written.

Description of Terms Used "Permuted Title Indexing" and "Keyword-in-Context Indexing" (KWIC) are synonymous and refer to the procedure of shifting sequences of words. The important word is printed in a designated keyword column while retaining its contextual significance through reproduction of surrounding words. "Discard words" are the obverse of important words or "keywords." "Discard words" simply are passed-by in the indexing procedures. Notation used herein for computer programs and construction









being relatively inexpensive when compared to stored indexing procedures. However, the ease of creating permuted title indexes is counter-balanced by the lack of any qualitative judgements as to the value of the word being indexed in relation to its possible importance to the user or researcher. The only determination made is whether the word is to be kept ( a keyword ) or to be discarded. This determination, while discretionary on the part of the programmer, is nonetheless completely quantitative in nature.

Origin of Permuted Title Indexes One of the earliest ventures in machine-generated-indexes was described to a meeting of The American Chemical Society in Atlantic City, New Jersey, on September 14, 1959, by Mr. Hans Peter Luhn of the International Business Machine Corporation.<sup>4</sup> It was on this occasion that the word KWIC was added to the computer field's dictionary of buzz words. (While often associated with IBM, KWIC is not a trademark of that company.) In the six years following the introduction of KWIC by Mr. Luhn, similarly constructed indexes have been generated in many segments of industry and education. Three examples of such applications are:

Chevron KWIC Index<sup>5</sup>

Journal of Association for Computing Machines<sup>6</sup>

Kansas Slavic Index<sup>7</sup>

The first is an index of articles published in The Journal of Petroleum Technology and The Society of Petroleum Engi-



neers Magazine of interest to the Chevron Oil Company.

The second example is an annual recapitulation of all articles in a trade publication. The third is in the Kansas State University and University of Kansas libraries as an ancillary aid to researchers in the field of Slavic writings.

These three examples serve to point up the wide range of subjects which may be served by permuted title indexes. While each example is somewhat different in format and construction, the techniques employed do not vary widely.





## Chapter II

### CONSIDERATIONS IN ESTABLISHING AN INFORMATION RETRIEVAL SYSTEM

General Requirements Any system to retrieve information must be designed to fit the problem at hand. This can be accomplished in one of two ways. First, the system may be designed from the beginning or secondly, a more general application may be modified to fit the problem. To attempt to design an information retrieval system which will apply to all categories (medicine, business, refining, engineering, etc.) is to presume that the end-user will have the requirement to know all things about all categories. This presumption is falacious with possibly the exception of one or two huge libraries which might serve as repositories for all the world's written works.

In view of this, the parameters of any problem in information retrieval may be reduced to three: 1) the data which is available 2) the processing system or systems which may be available to accomplish the task 3) the time dimension or urgency of the matter. The data parameter encompasses determination of file size (how much data should be indexed), conversion to a format which is compatible with the processing system to be used, assignment of weighting factors to distinguish the importance of one piece of data in relation to others being indexed (where appropriate), and constraints which are to apply to exception data (what field of endeavor is free of exceptions!).





The processing parameter is usually dictated by the available machine capability. Of course, a large indexing operation could require purchase or rental of equipment to perform the specific task at hand. In this case, computer hardware configurations having variable-word-length features, storage capacity to fit data collection, and high-access-speed to storage would play a large part in the selection of a computer to perform the task. The parameter of time is simply a pressure on the decision-makers responsible for the information handling project. Ample time allows for optimum data collection, machine selection, and output preparation. Stringent time requirements may adversely affect each element.

Additional Considerations Future expansion of data files imposes the requirement for elasticity in the construction of the initial indexing procedure. This is often called "open-file" construction. When the possibility exists that a particular data collection will grow over an extended period and that repeated indexing will be required (i.e. monthly, bi-monthly, etc.), system design must provide for the merging of new records with old. While much of the merging and sorting may be accomplished by machine, there remains the problem of how new data is to be merged--whether the criterion will be man-assigned codes or whether the computer is to be programmed to perform recognition chores. The latter is certainly desirable when the data may be recognized and stored without inclusion of a value



descriptor. However, when relative-judgements are required, man is still required to assign the "opinion" code.

The permuted-title-index method of expediting information retrieval is one step toward the day when it will be the function of the library to retrieve information, thereby freeing the researcher to spend his time digesting and evaluating the information in the field under study. "Research workers today, in danger of suffocation under the weight of published material, should not have to do the extra work of discovering what documents are actually relevant - searching the literature - unless they have some particular reason for wanting to do so." <sup>8</sup>

As the trend continues toward automated library service, information retrieval will be greatly facilitated, but new problems will arise. Specifically, the economics of running a large computer-oriented library solely for research are of a magnitude to provoke extensive debate, as the concept is not yet universally accepted. Coupled with the economic questions are the very real questions of who will operate this advanced system of high speed data retrieval. The thought of having the average user push buttons to operate a system with which the user is not totally familiar appears impractical and leads to the conclusion that trained information retrieval personnel will be required to render such an advanced system proper handling and efficiency of operation. <sup>9</sup>



Implications of Information Retrieval to Military Situations

"The real expense of information comes in not having it." <sup>10</sup> This quote actually comes from a trade publication describing efforts by the Goodyear Rubber Company to keep pace with the exploding volume of technical information pertinent to the rubber industry. However, the quote states a truism of military doctrine which over the years has been expressed in a variety of patriotic and political slogans -

"To be forewarned is to be forearmed"

"Know your adversary"

"In war, the winner spends too much,  
the vanquished spends too little"

Each of these expressions (and many more of similar intent) deals with knowledge of the opponent and presupposes a complete knowledge of the speaker's own internal operations.

Today the "expense" of missing information no longer can be calculated in terms of missing intelligence or poor scouting reports. The electronic, atomic, computerized defense establishment which this country has acquired for itself also demands a high degree of information gathering in terms of what inventories are available, where they are located, whether parts are interchangeable, what is the best delivery path, etc. Thus, the techniques for collecting data, indexing, publishing, and distributing large numbers of cross-reference manuals to end-users



around the world find increasing application in the military as well as civilian industry.





## Chapter III

### A PROBLEM IN PERMUTED TITLE INDEXING

Data A collection of Chemical Engineering M.S. and Ph.D. thesis titles with author names, school, and year of publication exists in the Department of Chemical and Petroleum Engineering. This collection, comprised of 3000 titles, has been assimilated from 144 universities in the United States and Canada and has been stored on punched cards. Each record consists of at least three punched cards and ranges to a maximum of six cards. The format of the input data is listed below.

1st card	Serial code (Columns 1-4)
	Author name (Columns 6-38)
	Year of publication (Columns 40-43)
2nd card	Title (Columns 7-55)
3rd card	Title continued (Columns 8-55) optional
4th card	Title continued (Columns 8-55) optional
5th card	Title continued (Columns 8-55) optional
Last card	School name (Columns 9-48)

Appendix B on page 61 shows a print-out of sample input data. The University of Kansas computer (IBM 7040-16K) was utilized to convert the data to printed format and to construct a permuted title index.

Method of Solution The problem is subdivided into six general sub-areas for discussion:

Record processing

Error detection



Keyword identification

Tape loading

Sorting

Write-out procedures

A general flow chart is shown on page 19 as Figure 2.

Record Processing A set of cards comprising a single record is read into the computer. Based on the pre-assigned format for each card in a set, the computer stores the information from each card, formulates the code (serial number of title, year of publication, school code number) which will be carried forward in each step of the program, searches for words, tests the words found against a dictionary of discard words, and records the information on three tapes: a master record tape, a keyword tape, and an author tape.

This procedure is repeated until the last card, which is coded "last card," is read. The main-line computer program is listed in Appendix A, pages 33 through 44.

Error Detection As each card is read, its format is compared with the pre-assigned formats for each card in the set, and, if an exception exists, the exception card is printed out with a message indicating whether the error exists in author, title, or school card. Corrective action is then taken manually by repunching the data card which has been found to be incompatible with the specified format.

Spelling errors are not detected by this procedure and, in fact, are ignored in the preparation of the per-



muted index. One exception to the spelling error rule exists when formulating the school code number. A list of all schools applicable to the problem is stored in the computer. The school data card information is compared to this list until an exact comparison exists. The code corresponding to the school, as listed in Appendix C, page 62, is then assigned. Should an exact comparison not exist, an error message will be printed showing the school data card which is incompatible with the school dictionary file. If the school simply is not listed, an additional entry must be added to the school dictionary. If the school data card is misspelled, repunching is required.

Keyword Identification When a record has been determined to be in proper format, the title cards are searched character by character. Any combination of letters with a blank character at both ends of the combination is considered a possible keyword which is then compared to the words in the discard dictionary. If the combination of characters being tested corresponds with a word in the discard dictionary, no further action is taken and the program proceeds to locate the next succeeding group of characters in the title cards. If the combination of characters does not coincide with any discard word, a keyword is determined and is stored on the tape containing keywords, location of the first character of the keyword in relation to the full title, and the serial code. This procedure is repeated on each of the four title data arrays until two characters in succession are deter-





mined to be blank. Thus, if a complete title exists on a single input data card, the program will sense the end of the first card when 1) the column count reaches the limit prescribed for the card being tested or 2) two successive blanks are determined. At this point, the program will check in succession the first column of each of the remaining title data arrays to determine the existence or non-existence of additional title information. If non-existence is determined in each remaining array, end of title is confirmed and the program proceeds to the next element.

Tape Loading Three tapes are loaded with data to be used in printing output listings. First, a master bibliographic tape is prepared which contains serial code, full title, and author as a single entry. Second, a keyword tape is prepared. Each keyword which is determined is loaded as a single entry on this tape. Serial code and keyword location are carried along as mentioned above. Third, an author tape is prepared consisting of serial code and author name.

Sorting The master bibliographic tape is loaded in order by serial code and no sorting is performed on this tape. The keyword tape is sorted alphabetically using the IBM Generalized Sort System <sup>11</sup> for the 7040 Computer. This sort program is shown in Appendix A, page 59. The author tape was not sorted due to the fact that serial codes are assigned alphabetically by author name. However, if the





file were increased and serial codes were assigned on a basis other than alphabetic-author, this tape is so constructed that it may be sorted by the same sorting program that is used to sort the keyword tape.

Write-out Procedures Separate write-out programs for each output tape are listed in Appendix A on pages 50, 52 , and 56. The programs for the master bibliographic tape and author tape are straightforward and involve only read-write operations within the computer.

The program for writing the KWIC index requires both the sorted keyword tape and master bibliographic tape as input data to the computer. The keyword tape is read, the master tape is searched to find the bibliographic listing corresponding to the code applicable to the keyword being listed, the title is positioned in a holding array within the computer by the keyword location, and the permuted title is then written out.

When a keyword is located near the end of a lengthy title, much of the meaning of the title may be lost due to truncation of the title. In this event, provisions are included in the write-out program for the keyword tape to "wrap-around" the first forty characters of the title.

Description of Programs The computer instructions for the problem are divided into five independent segments:

- 1- Main program to process data and load tapes  
(Appendix A, page 33)
- 2- Sorting instructions  
(Appendix A, page 59)



- 3- Write-out of master bibliographic list  
(Appendix A, page 50)
- 4- Write-out of permuted title index  
(Appendix A, page 52)
- 5- Write-out of author list  
(Appendix A, page 56)

The Main computer program has the basic functions of checking and converting. Checking: Each data card must be checked for correctness of the format, and notice of error must be returned to the programmer by a computer generated error message. Converting: Once the data has been transferred from the storage medium of punched cards to the medium of electronic core storage, the physical restrictions of record size (i.e. eighty columns of punched data per card) are removed, and records of greater length than eighty characters may be established in the computer. As keywords are identified, the information is transferred from one storage medium (core) to another (tape). Figure 1 at the top of the next page describes the final format of the data as it is transferred from the computer and stored on tape.

Four subprograms are utilized in connection with the main program:

- 1- The NBRT function subprogram is used to determine whether a character being tested is numeric or non-numeric. The answer returned to the main program is either true or false. (True if character under test is a number; false if not a number).



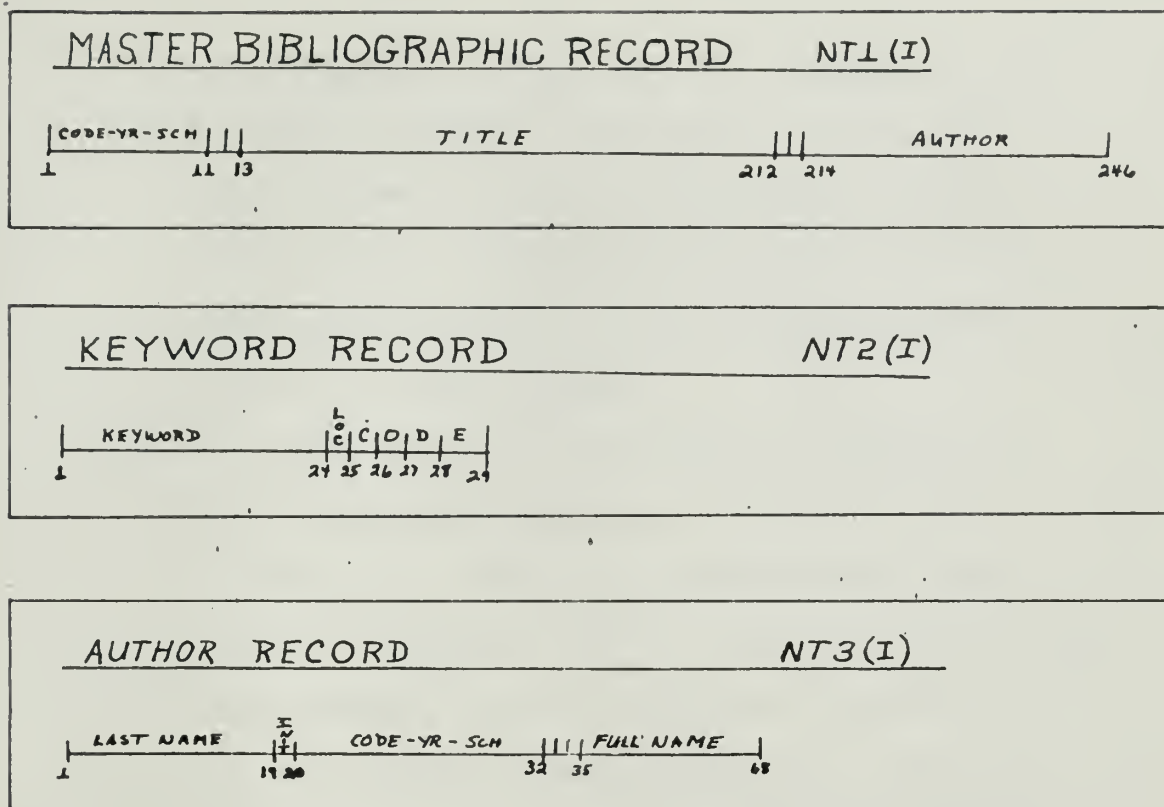


FIGURE 1

- 2- The PACK subprogram, which is written in the Macro Assembly Program (MAP), is used to speed-up the internal search procedure which determines school-code numbers. This subprogram condenses or PACKS characters which are stored one character per computer word to a maximum of six characters per computer word. Thus, the school-data-card information which is read into the computer on a character by character basis may be converted to allow faster comparison to school dictionary which is read and stored



six alphabetic characters per computer word.

- 3- The KSKOL function subprogram is used in the formulation of the serial-year-school code number. The program searches the school dictionary and returns the school code number to the main program.
- 4- The KICK subroutine subprogram performs the decision-making function of whether a group of characters under test constitutes a discard word or a keyword. If the group of characters being tested does not correspond exactly to any word in the discard dictionary, the word is returned to the main program for continued processing. Otherwise, a zero signal is returned to the main program to initiate search for the next group of characters.

The second segment of computer instructions is the Sorting instructions. The tape containing the information to be sorted is mounted on the tape unit specified by the programmer, and the control cards for the IBM Generalized Sort System are read into the computer. When sorting is complete, the sorted tape will be located on the tape-drive assembly selected by the computer - the identification of which will be listed on the computer console typewriter record.

The third, fourth, and fifth segments of the instructions are programs to list or print-out the information





stored on the three tapes involved. Each write-out program requires its corresponding tape to be mounted on a specified tape drive assembly as input data to the write-out program. The write-out program for the permuted title index also requires the master bibliographic tape as input data as mentioned above.



# GENERALIZED FLOW CHART

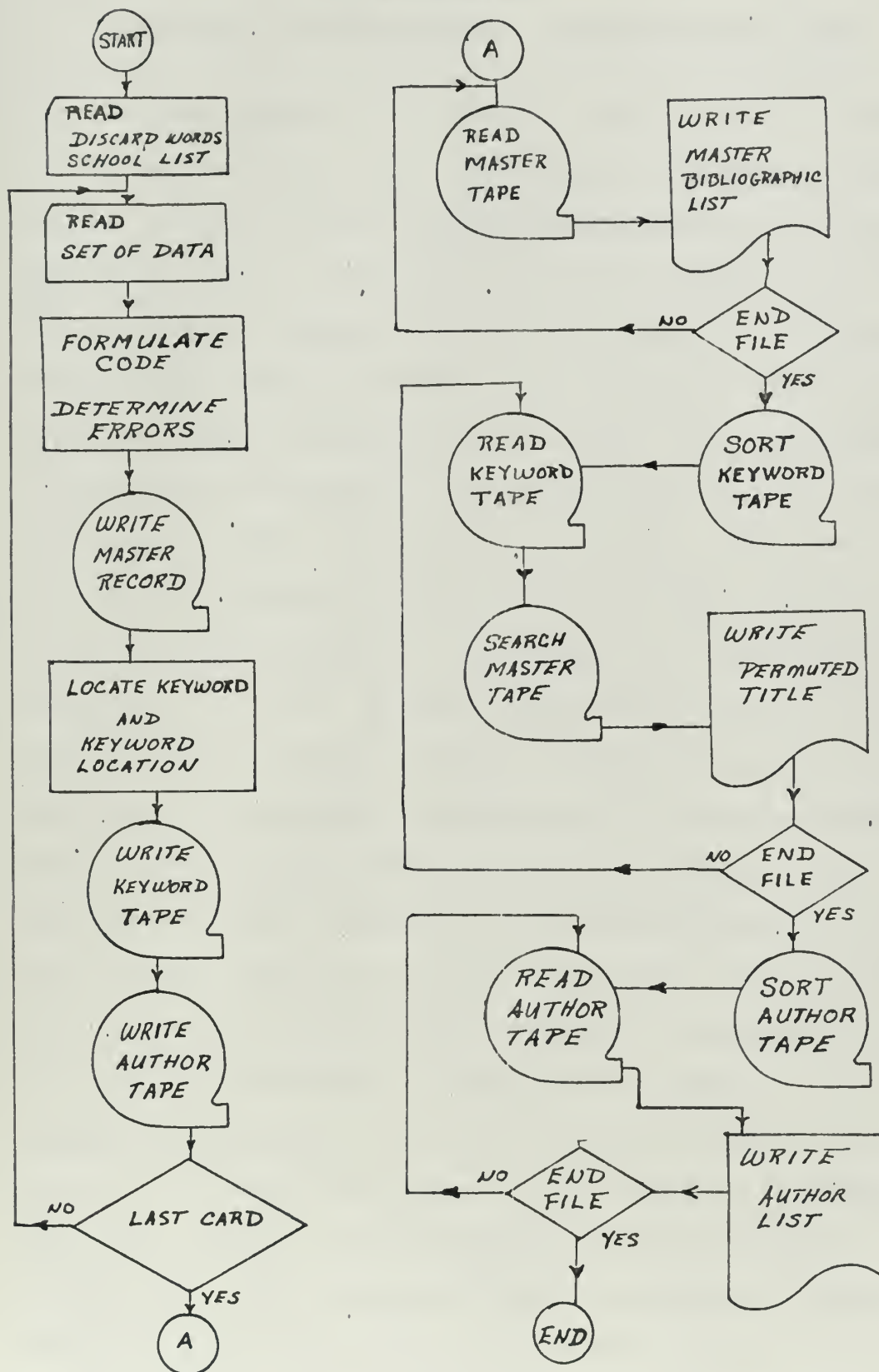


FIGURE 2



## Chapter IV

### ANALYSIS OF PROCEDURES AND COMPUTER TECHNIQUES

The key element in creating a permuted title index is the keyword file. This file must be capable of being sorted and must contain the applicable master title or reference code to facilitate search for the master title. The preferable method of obtaining titles is to include them in the keyword records. In this manner, titles are "carried along" as keywords are sorted. This procedure allows direct write-out of the sorted tape and obviates the need to search the master record file. This procedure was attempted in the solution of this problem and was successfully used on a test data deck consisting of sixty sets of data. However, complications arose when the full three thousand record sets were processed. The keyword tape file required six tape reels to store all keywords found plus the applicable master record. This fact rendered the solution to the problem impossible with the computer equipment at hand because the generalized sort system for the University of Kansas computer will sort only to a maximum of three reels of tape.

Alternate Strategies The need to reduce the size of the keyword file being apparent, it was decided to eliminate the master listing carried along with the keyword and serial code. This change allowed a reduction in record length from 246 words per record to 29 words per record.

This change, in reducing the keyword tape by a factor



of eight, did circumvent the physical limitation imposed on the preferred solution. But, and this is an important but, the solution became dependent on time consuming tape-search procedures.

Another alternative, which was not tried but which now appears most sensible, would be to compress the keyword file by a factor of six. This could be accomplished through the use of the PACK subprogram (page 58) which is used in the school-code determination program. By condensing the keyword file, slight modification would be required to be made to field specifications in the SORT control cards, but this would pose no problem. Once the keyword tape was sorted, write-out procedures could be executed with one small addition to the write-out program. The compressed words on the keyword tape must be de-compressed or UNPACKED in order that character by character control may be exercised in the positioning of the permuted title. The unpacking operation could be performed through the use of the subprogram UNPACK which is listed in Appendix A, page 58 .

Utilization of Computer Time Because of the need to use the tape storage media rather than internal computer memory to hold the master file, keyword file, and author file, much of the advantage gained in using a high speed computer is blunted by a repeated tape-READ and tape-WRITE procedures which are inherently slower operations than the movement of data within core memory. The program presented with





this paper essentially deals with one record at a time, and no analysis is made to determine the optimum use of core storage to speed the computer operations. The matter of computer time usage is an economic one which must be taken into consideration in weighing the cost of generating permuted title indexes versus the expected gain in research productivity.

Some methods for optimizing the use of computer time are discussed below as applied to the problem at hand. First, and most importantly, the strategy of tape-searching must be avoided if possible because each increase in file size requires greater time to locate each title record. The need to "carry along" the title as keywords are sorted is of paramount importance when using a tape-oriented system. Disk storage, with its high access speed, would lessen the dilemma if sufficient disk space were available to hold the complete files stored on tape in the tape-oriented system.

If all avenues of escape are closed and tape-search procedures remain as the only solution to the problem, much time may still be saved by incorporating a dual master file. This procedure involves duplicate master file tapes. The program is written to search each tape on an alternating basis so that when one search is complete, and the tape is rewinding, the program may proceed to search the alternate tape without waiting for the completion of rewind procedures.

Secondly, the keyword file itself may be reduced in size by increasing the size of the discard dictionary.



This solution will reduce sort and write-out time, but it will require longer running time in loading the tapes.

The fifteen common words, in order of occurrence, which are listed below account for 30% of all words in titles. <sup>12</sup>

1. of	6. by	11. to
2. the	7. a	12. some
3. and	8. with	13. as
4. in	9. for	14. at
5. on	10. from	15. an

These fifteen words are the work horses of the discard dictionary. The incorporation of additional discard words in the dictionary provides for neater output listing, but the increase requires much additional computer running time. On the test data, increasing the discard dictionary from fifteen to thirty words caused running time to increase in approximately the same ratio (1-2).

Thus, in designing an optimum system, some balance must be reached between increases in discard dictionary size and elimination of low-value words in the permuted index. The list of discard words for this problem is shown in Appendix A, page 60.

The third area of discussion pertains to READ-WRITE operations. At the completion of program de-bugging, remaining unused core space should be divided into two holding arrays. This would allow READ-in of more than one data set per READ command, and more than one WRITE-output per WRITE command. By reducing the number of referrals to peripheral equipment from the main computer, additional



time would be conserved.

Interpretation of Output Appendix C, fold-out page 62, shows samples of three elements of the permuted title index. On the left is the school code dictionary. The center section shows the permuted index, and the right hand section, the master bibliographic listing. The school dictionary is in numerical order; the permuted index is in alphabetical order by keyword; and the master listing is in order of serial code number. To obtain the complete title which pertains to a keyword of interest to the researcher, the serial code serves as reference to enter the master listing. The school of publication may be read directly using the school code number which is part of the serial-year-school code number.



## Chapter V

### CONCIUSION

Two questions were posed at the outset of this problem:

- 1- Did a feasible solution to the problem of permuted title indexing exist when a fixed-word-length computer was used rather than the apparently more logical variable-word-length computer?
- 2- Given that a fixed-word-length computer would be used in the solution of the problem in permuted title indexing, did a feasible solution exist with the use of the engineering-oriented computer language FORTRAN?

The answers to these two questions are in the affirmative, but with some qualifications.

#### Summation of Findings

- 1- Permuted title index problems may be solved using fixed-word-length computers.
- 2- FORTRAN computer language is adequate to perform the tasks involved in creating a permuted index. Variable format capability, while not a mandatory requirement, would have simplified the WRITE portions of the overall program.
- 3- Discard dictionary size should be kept as small as practicable.





- 4- Keyword files which will be sorted by keyword should contain full title if at all possible. The procedure of tape-searching should be avoided when other solutions are available.

Projections of the Future This project involved the classic problems of creating a permuted title index. As machine capabilities increase, as programming languages incorporate new concepts applicable to information retrieval, and as the need for more detailed research tools become apparent, it will be necessary to improve the permuted index from a simple alphabetic listing to a more detailed listing involving hierarchies of information. This will require establishment of new and different criteria for sorting.

This type of information retrieval tool has some of its most promising applications in the field of engineering, both practical and theoretical.

The question remains whether engineers will provide their own research tools or wait for the librarians to do it for them.

Food for Thought "When computers have been programmed to nose out more new mathematical proofs, when one-pack patiences have been mechanically played to stimulate the nature of a control system, when a computer is used, but reliably (as it will be), to paint pictures and write poems, and equipped with a machine-sized thesaurus to translate and therefore comparatively to identify differences of



context in metaphysical and theological statements in different languages, when all this happens, will the programmer, the analytical wielder of this new mathematical paintbrush, be an artist, or will he be a scientist?" <sup>13</sup>



## REFERENCES

1. 7040-7044 Fortran Programmer's Guide, IBM Operating System Manual No. C28-6329-1.
2. Macro Assembly Program (MAP), IBM Operating System Manual No. C28-6335-2.
3. Howerton, Paul W., Information Handling, 1963.
4. Luhn, Hans Peter, Keyword-in-Context for Technical Literature (KWIC Index), 1959.
5. Chevron KWIC Index, June, 1965.
6. Journal of the Association for Computing Machinery, October, 1963.
7. Kansas Slavic Index, 1963.
8. Fosket, D.J., "Freeing the Mind," London Times Literary Supplement, April 27, 1962, page 10.
9. Luhn, Hans Peter, "Automated Intelligence Systems," Information Retrieval Management, 1962, page 97.
10. "Goodyear Automates Fact Finding," Rubber World, Volume 146 #4, July, 1962, as reprinted in Readings in Information Retrieval, 1964.
11. 7040-7044 Generalized Sorting System, IBM Operating System Manual No. C28-6337-3.
12. Keyword-in-Context (KWIC) Indexing Program for the IBM 1401 Data Processing System, Application Program Bulletin No. H20-0348.
13. Masterson, Margaret, "The Intellect's New Eye," London Times Literary Supplement, April 27, 1962.



## BIBLIOGRAPHY

1. The Association for Computing Machinery, Journal of the Association for Computing Machinery, New York, New York, October, 1963.
2. Bell, D.A., Information Theory and Engineering Applications, London: Pitman Press, 1962.
3. Chevron Oil Company, Chevron KWIC Index, June, 1965.
4. Dun and Bradstreet, Dun's Review, New York, New York, September, 1965.
5. Howerton, Paul W., Information Handling, London: Spartan Books, 1963.
6. IBM Corporation, 7040-7044 Fortran Programmer's Guide, IBM Operating System Manual No. C28-6329-1, New York, New York.
7. IBM Corporation, 7040-7044 Generalized Sorting System, IBM Operating System Manual No. C28-6337-3, New York, New York.
8. IBM Corporation, Keyword-in-Context Indexing, IBM Manual E20-8091, New York, New York.
9. IBM Corporation, Keyword-in-Context (KWIC) Indexing Program for the IBM 1401 Data Processing System, Application Program Bulletin H20-0348, New York, New York.
10. IBM Corporation, Macro Assembly Program (MAP), IBM Operating System Manual No. C28-6335-2, New York, New York.





11. Licklider, J.C., Libraries of the Future, Cambridge, Massachusetts: MIT Press, 1965.
12. London Times Press, London Times Literary Supplement, London, April, 1962.
13. Luhn, Hans Peter, "Automated Intelligence Systems," Information Retrieval Management, American Data Processing, 1962.
14. Luhn, Hans Peter, Keyword-in-Context Index for Technical Literature (KWIC Index), Yorktown Heights, New York: IBM, 1959.
15. Perry, J.W., A. Kent, and M.M. Berry, Machine Literature Searching, Cleveland, Ohio: Science Publishers Inc., 1956.
16. Sharp, Harold S., Readings in Information Retrieval, New York: Scarecrow Press, 1964.
17. The University of Kansas Libraries, Kansas Slavic Index, Lawrence, Kansas, 1963.
18. Vicery, B.C., On Retrieval System Theory, London: Butterworths, 1961.
19. Western Reserve University, Information Retrieval in Action, University Heights, Ohio, 1963.



## FORTRAN NOTATION

KD(I)	DATA CARD- (I) CONTROLS COLUMN IDENTIFICATION
KDCON1(I)	AUTHOR CARD DATA
KDCON2(I)	TITLE CARD DATA
KDCON3(I)	TITLE CARD DATA (CONT)
KDCON4(I)	TITLE CARD DATA (CONT)
KDCON5(I)	TITLE CARD DATA (CONT)
KDCONS(I)	SCHOOL CARD DATA
SKL(I,J)	SCHOOL DICTIONARY (I) CONTROLS SCHOOL (J) CONTROLS WORDS IN SCHOOL NAME
L(I)	AN ARRAY CONTAINING ALL KEYPUNCH REPRESENTATIONS IN ALPHANUMERIC MODE
TMPKW(I)	TEMPORARY KEYWORD USED IN TESTING PROCESS
DCD(I,J)	DISCARD DICTIONARY (I) CONTROLS WORDS (J) CONTROLS LETTERS
KW(I)	KEYWORD AS DETERMINED BY SUBROUTINE KICK
NT1(I)	RECORD AS COMPILED IN FINAL FORM FOR STORAGE ON MASTER TAPE
NT2(I)	KEYWORD RECORD AS COMPILED IN FINAL FORM FOR STORAGE ON KEYWORD TAPE
NT3(I)	AUTHOR RECORD AS COMPILED IN FINAL FORM FOR STORAGE ON AUTHOR TAPE
NTEST(I)	HOLDING ARRAY USED IN DETERMINATION OF WORDS
KOUNT	CARD COUNT INDICATOR TO DETERMINE WHICH CARD IN SET IS BEING OPERATED UPON
NTAPE1	TAPE NAME FOR MASTER BIBLIOGRAPHIC LISTING
NTAPE2	TAPE NAME FOR KEYWORD TAPE
NTAPE3	TAPE NAME FOR AUTHOR TAPE
NC1(I,J)	HOLDING ARRAY FOR KEYWORDS AND LOCATION USED TO FACILITATE LISTING OF KEYWORDS
NRITE(I)	HOLDING ARRAY USED FOR POSITIONING TITLE IN ORDER THAT KEYWORD WILL BE POSITIONED IN PROPER COLUMN



NST ----- MAINLINE COUNTER TO SPECIFY WHICH COLUMN OF  
NT1 BETWEEN COLS 13 AND 212 IS BEING  
OPERATED UPON

LOC ----- MAINLINE COUNTER WHICH DETERMINES RELATIVE  
LOCATION BETWEEN 1 AND 200 OF FIRST LETTER OF  
KEYWORD

LLOC ----- WRITE-OUT PROGRAM FOR KEYWORD TAPE - RELATIVE  
LOCATION OF FIRST LETTER OF KEYWORD IN  
NRITE (1 THRU 600)

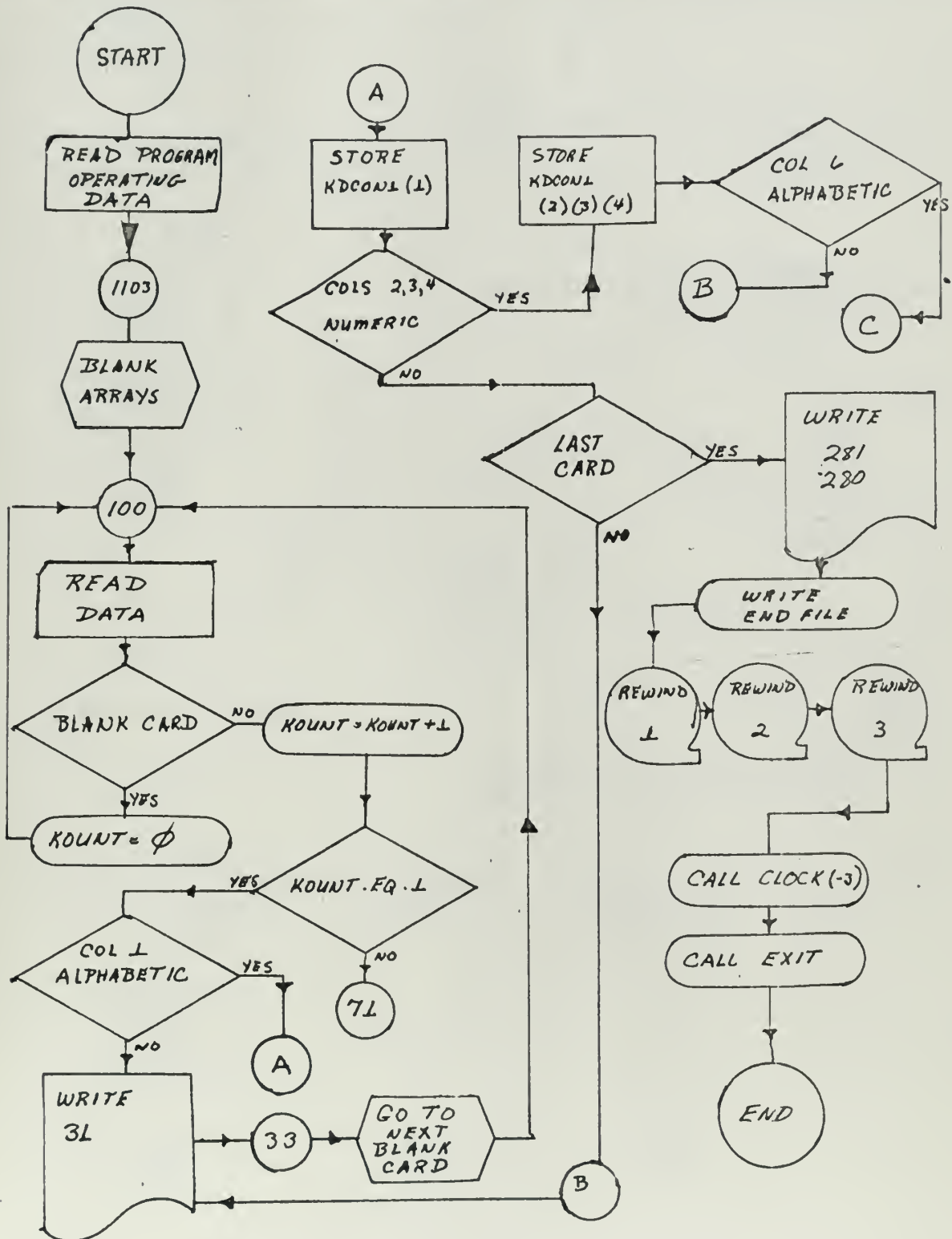
L1 ----- LOWER LIMIT OF NRITE(I) WHICH IS TO BE  
WRITTEN OUT

L2 ----- UPPER LIMIT OF NRITE(I) WHICH IS TO BE  
WRITTEN OUT



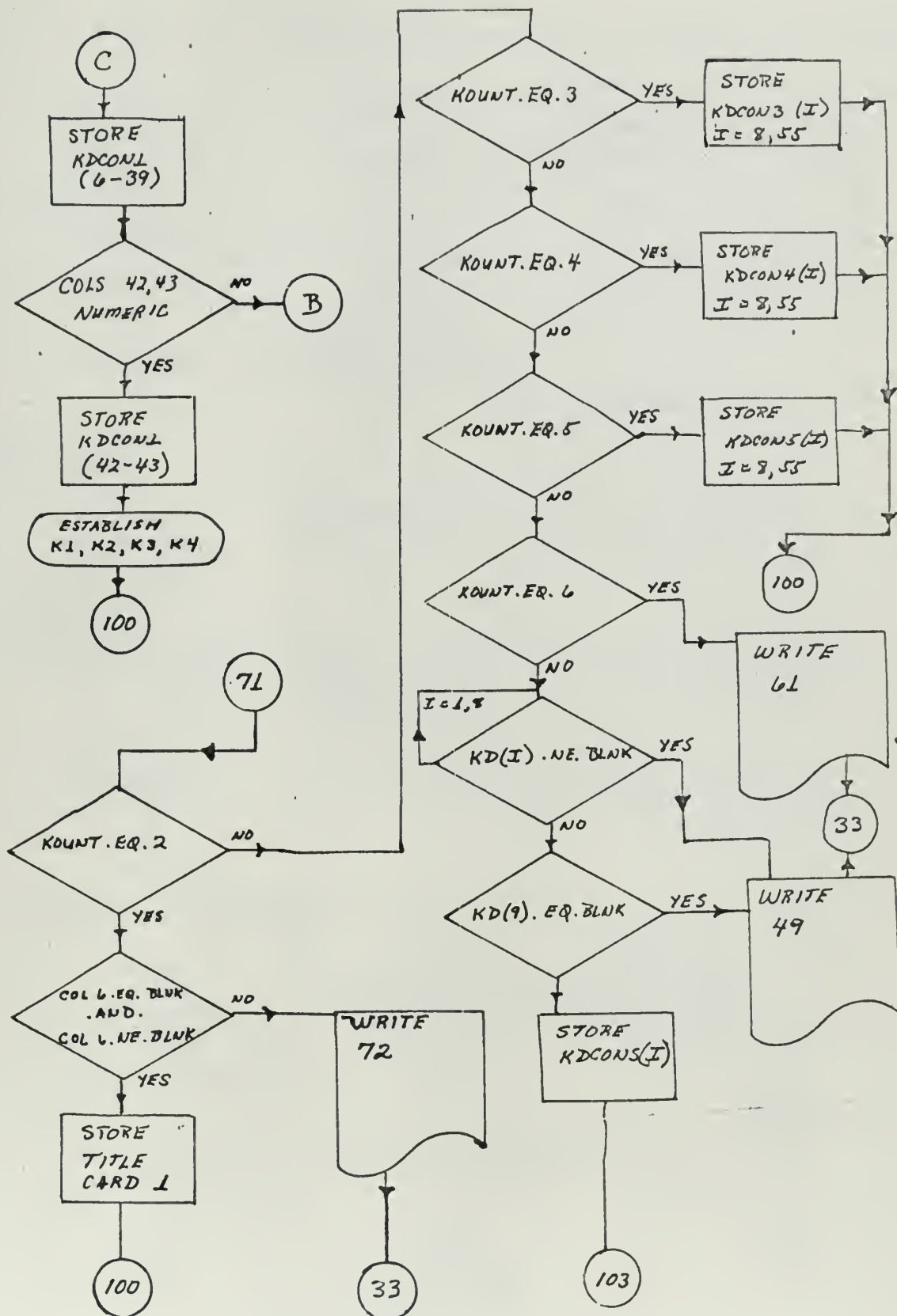


## FLOW CHART FOR MAINLINE PROGRAM TO LOAD TAPES

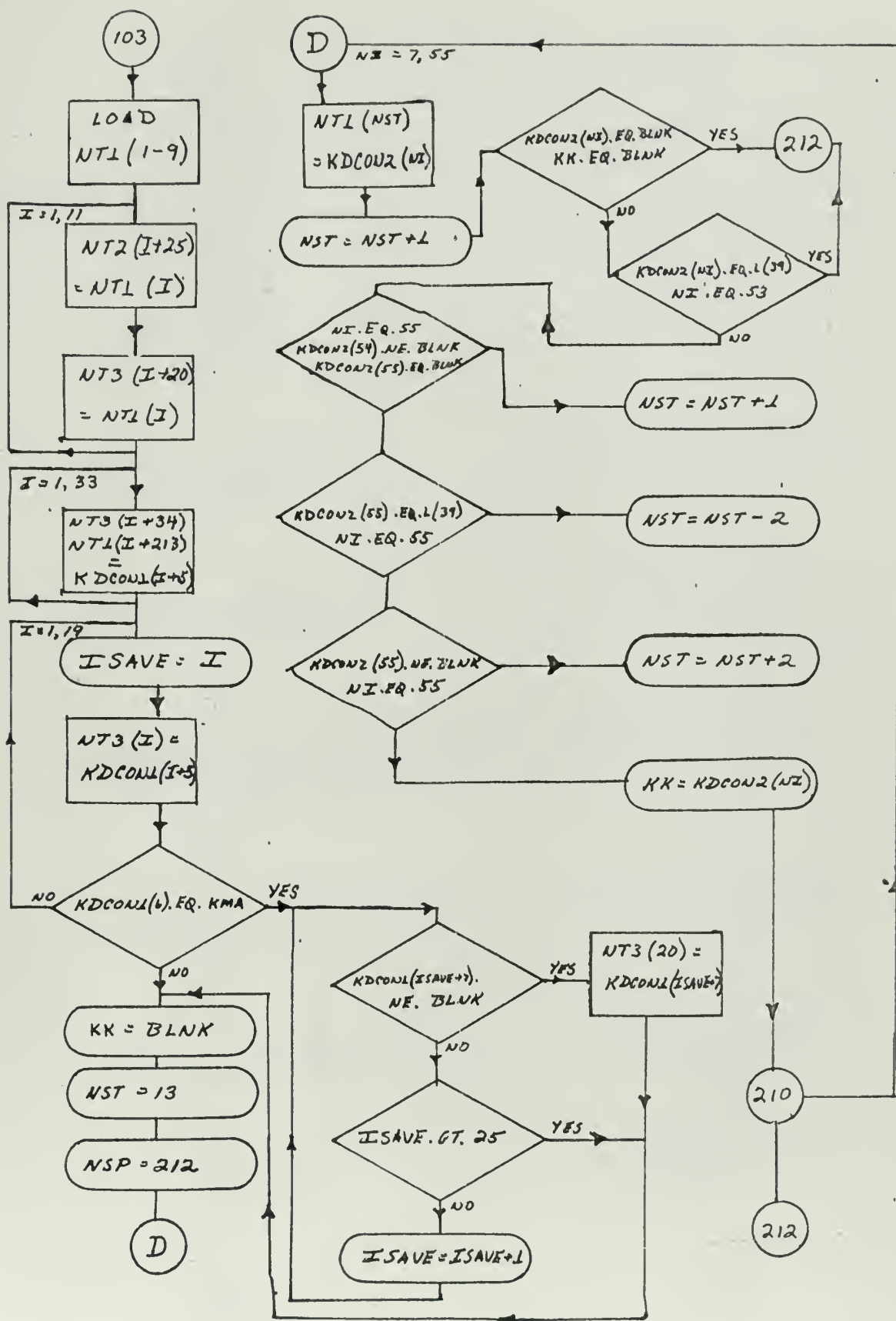




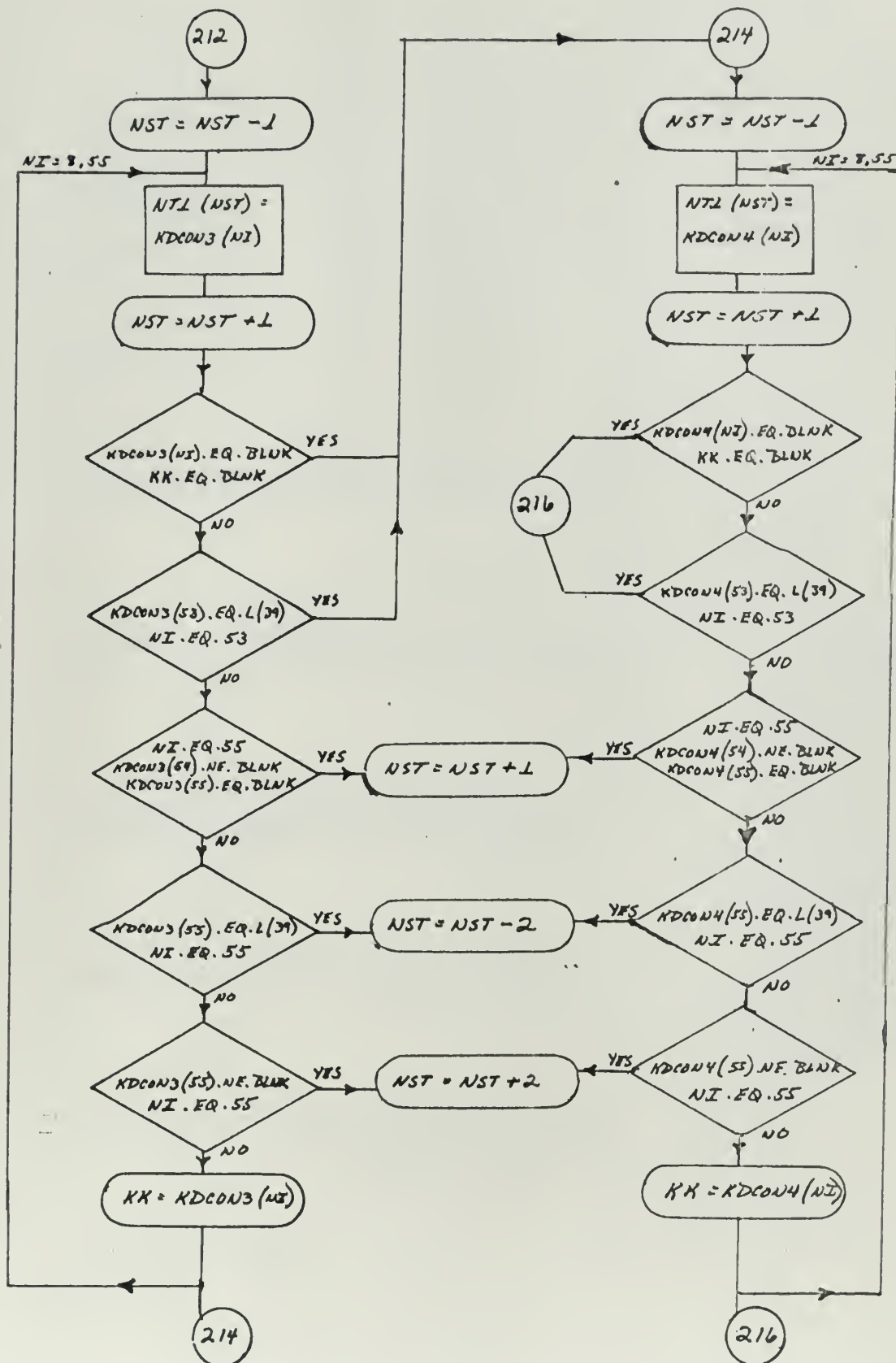




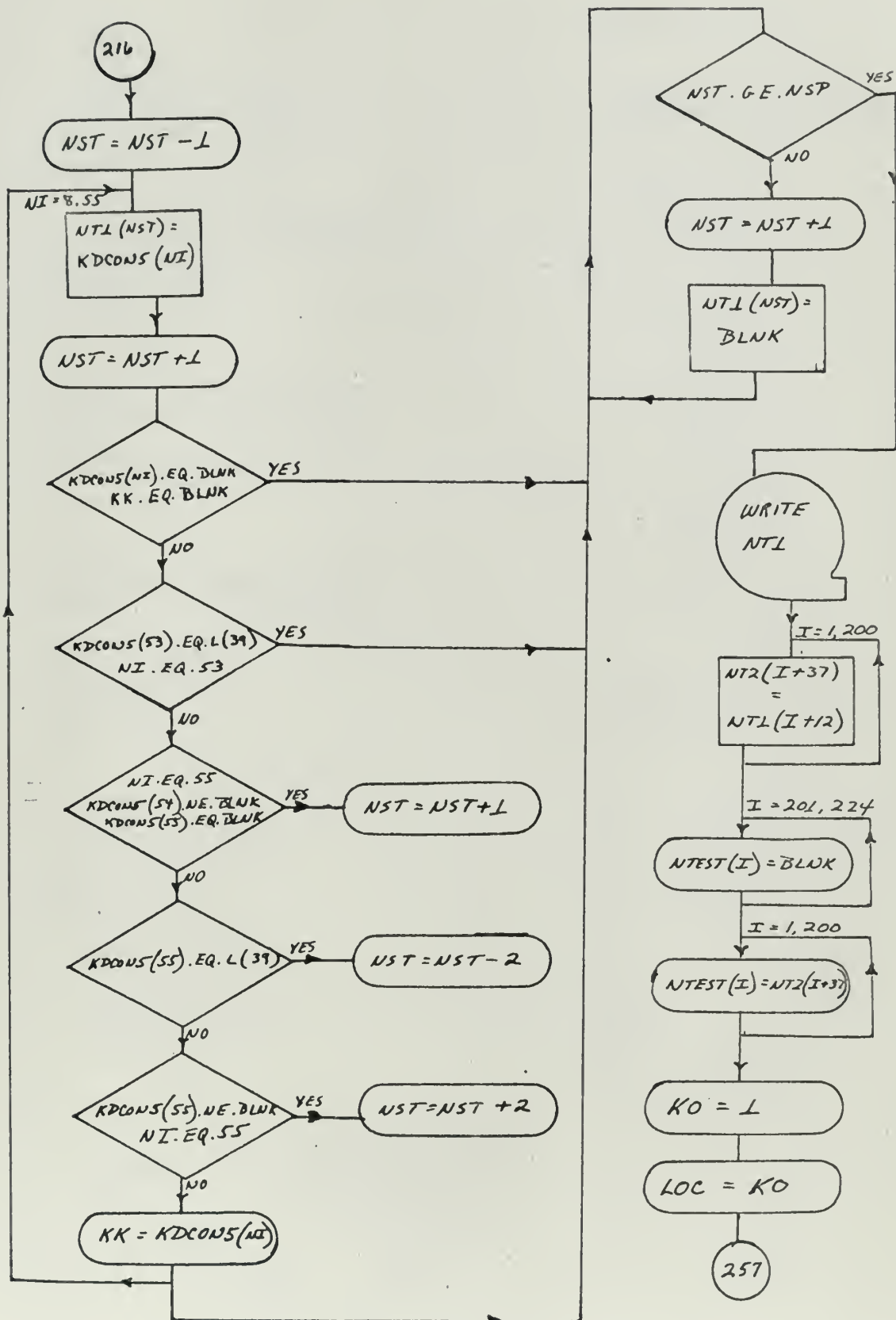






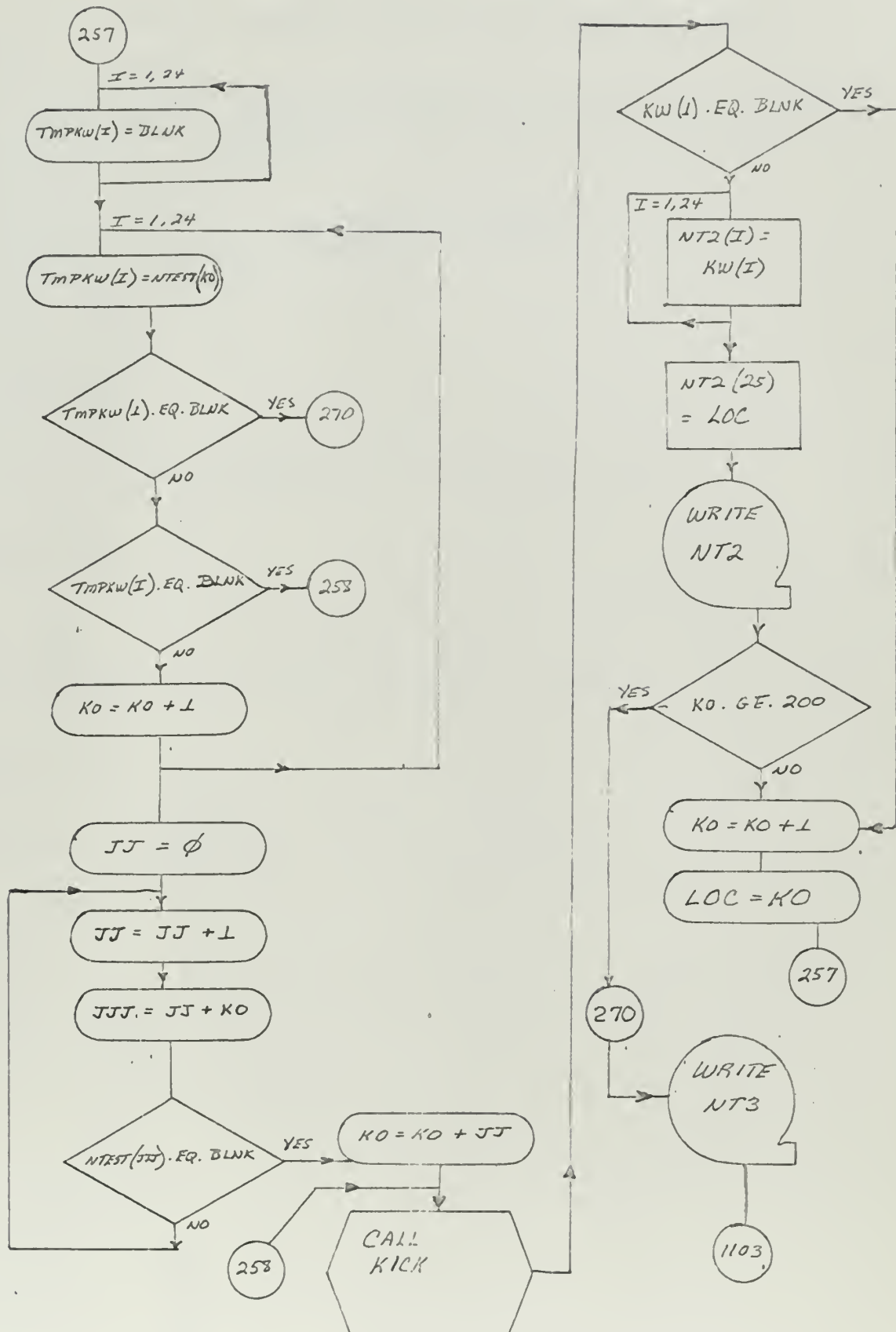














## SOURCE STATEMENT

1 FORMAT ( 80A1)

CC 2 I = 1,80

2 IF (KD(I) .NE. BLNK ) GO TO 4

3 KOUNT = 0

GC TC 100

4 KCOUNT = KCOUNT + 1

IF (KCOUNT .NE.1 ) GO TO 71

IF(NERT(KD(1))) GO TO 31

KDCON1(1) = KC(1)

CC 5 I=2,4

IF(.NOT. NERT(KD(I))) GO TO 35

5 KDCON1(I) = KC(I)

IF(NERT(KD(6))) GO TO 31

CC 6 I = 6,39

KDCON1(I) = KC(I)

6 CONTINUE

CC 7 I = 42,43

IF (.NOT.NERT(KD(I))) GO TO 31

7 KDCON1(I) = KC(I)

C CREATION OF TEMPORARY REFERENCE NUMBER

K1 = KC(1)

K2 = KC(2)

K3 = KC(3)

K4 = KC(4)

GC TC 100

C ROUTINE FOR SECOND CARD IN SET

71 IF (KOUNT.NE.2) GO TO 41

IF(KD(6).EQ.BLNK.AND.KD(7).NE.BLNK ) GO TO 73

WRITE (6,72) K1,K2,K3,K4

72 FORMAT (10X,34FERROR IN TITLE CARD - LAST SERIAL , 4A1)

GO TO 33

73 CONTINUE

CC 74 I = 7,55

74 KDCON2(I) = KC(I)

GO TO 100

C ROUTINE FOR CARDS 3,4,5,6

41 CONTINUE

44 IF((KD(7).EQ.BLNK).AND.(KD(8).NE.BLNK).AND.(KOUNT.EQ.3))GO TC 42

IF((KD(7).EQ.BLNK).AND.(KD(8).NE.BLNK).AND.(KOUNT.EQ.4))GO TC 52

IF((KD(7).EQ.BLNK).AND.(KD(8).NE.BLNK).AND.(KOUNT.EQ.5))GO TC 55

IF (KOUNT.GT.6) GO TO 58

47 DO 45 I = 1,8

45 IF(KD(I).NE.BLNK) GC TO 48

IF(KD(9).EQ.BLNK) GC TO 48

CC 46 I = 9,55

46 KDCON3(I) = KC(I)

GC TC 103

42 CC 43 I = 8,55

43 KDCON3(I) = KC(I)

GO TO 100

C 48 WRITE (6,49) K1,K2,K3,K4



49 FORMAT (10X,50HERRCR IN TITLE CARD OR SCHOOL CARD - LAST SERIAL =  
1, 4A1)

C -----

GO TC 33  
52 DO 53 I = 8,55  
53 KCCCN4(I) = KD(I)  
GO TC 100  
55 DO 56 I = 8,55  
56 KCCCN5(I) = KD(I)  
GO TC 100  
58 WRITE (6,61) K1,K2,K3,K4  
61 FORMAT(10X,37HCARD COUNT EXCEEDS 6 - LAST SERIAL = , 4A1)  
GO TC 33  
59 CONTINUE

C NOW WE LOAD ALL ARRAYS IN CONDENSED FORM

GO TC 103

C -----

C ERROR STATEMENT FOR AUTHOR CARD

31 WRITE (6,32) KC  
32 FORMAT ( 10X,17HAUTHCR CARD ERROR, 10X, 80A1)  
33 READ (5,1) KD  
DO 34 I = 1,80  
34 IF ( KD(I).NE.BLNK) GO TO 33  
KOUNT = 0  
GO TC 100

C -----

C TEST FOR LAST CARD

35 DO 36 I = 11,80  
36 IF (KD(I) .NE.L(9)) GO TO 31

C END OF FILE SIGN -- 24 LETTER Z=S

DO 280 I= 1,24  
NT1(I) = L(36)  
NT2(I) = L(36)  
280 NT3(I) = L(36)  
WRITE (NTAPE1) (NT1(J), J= 1,246)  
WRITE (NTAPE2) (NT2(J), J=1,29)  
C WRITE (NTAPE3) (NT3(J), J=1,68)  
GO TC 1000

C -----

C PROGRAM TO LOAD TAPES 1,2,+3

103 CONTINUE  
DO 201 I=1,246  
NT1(I) = BLNK  
201 NT2(I) = BLNK  
DO 202 I= 1,48  
202 NT3(I) = BLNK  
KSCHL = KSKOL(KDCONS)  
J1 = KCCCN1( 1)  
J2 = KCCCN1( 2)  
J3 = KCCCN1( 3)  
J4 = KCCCN1( 4)  
J6 = KCCCN1(42)  
J7 = KCCCN1(43)  
NT1(1) = J1  
NT1(2) = J2





## SOURCE STATEMENT

```

      NT1(3)  = J3
      NT1(4)  = J4
      NT1(6)  = J6
      NT1(7)  = J7
      NT1(9)  = KSCHL
      DO 203 I=1,11
      NT2(I+25) = NT1(I)
203  NT3(I+20) = NT1(I)
      DO 204 I = 1,33
      NT3(I+34) = KCCCN1(I+5)
204  NT1(I+213) = KCCCN1(I+5)
C   LCAD AUTHCR NAME AND FIRST LETTER FOR SORTING
      DO 230 I=1,19
      ISAVE = I
      NT3(I) = KCCCN1(I+5)
      IF (KCCCN1(I+6).EQ.KMA)   GO TO 231
230  CONTINUE
      GO TO 239
232  CONTINUE
231  IF (KCCCN1(ISAVE+7).NE.BLNK)   GO TO 238
      IF (ISAVE.GT.25)   GO TO 239
      ISAVE = ISAVE +1
      GO TO 232
238  NT3(20) = KCCCN1(ISAVE+7)
239  CONTINUE
C   -----
C   NOW SEARCH FOR KEYWORD AND KW LOCATION
C   -----
      KK = BLNK
      NST = 13
      NSP = 212
      DO 210 NI= 7,55
      NT1(NST) = KCCCN2(NI)
      NST = NST+1
      IF ((KCCCN2(NI).EQ.BLNK).AND.(KK.EQ.BLNK))   GO TO 212
      IF ((KCCCN2(53).EQ.L(39)).AND.(NI.EQ.53))   GO TO 212
      IF ((KCCCN2(54).NE.BLNK).AND.(KCCCN2(55).EQ.BLNK).AND.(NI.EQ.55))
1  NST = NST +1
      IF ((KCCCN2(55).EQ. L(39)) .AND.(NI.EQ.55))   NST = NST -2
      IF ((KCCCN2(55).NE.BLNK) .AND.(NI.EQ.55))   NST = NST +2
      KK = KCCCN2(NI)
210  CONTINUE
212  CONTINUE
      NST = NST - 1
      DO 213 NI = 8,55
      NT1(NST) = KCCCN3(NI)
      NST = NST +1
      IF ((KCCCN3(NI).EQ.BLNK).AND.(KK.EQ.BLNK))   GO TO 214
      IF ((KCCCN3(53).EQ.L(39)).AND.(NI.EQ.53))   GO TO 214
      IF ((KCCCN3(54).NE.BLNK).AND.(KCCCN3(55).EQ.BLNK).AND.(NI.EQ.55))
1  NST = NST +1
      IF ((KCCCN3(55).EQ. L(39)) .AND.(NI.EQ.55))   NST = NST -2
      IF ((KCCCN3(55).NE.BLNK) .AND.(NI.EQ.55))   NST = NST +2
      KK = KCCCN3(NI)
213  CONTINUE

```





214 CONTINUE

NST = NST - 1

DO 215 NI = 8,55

NT1(NST) = KCCCN4(NI)

NST = NST + 1

IF ((KCCCN4(NI).EQ.BLNK).AND.(KK.EQ.BLNK)) GO TO 216

IF ((KCCCN4(53).EQ.L(39)).AND.(NI.EQ.53)) GO TO 216

IF ((KCCCN4(54).NE.BLNK).AND.(KCCCN4(55).EQ.BLNK).AND.(NI.EQ.55))

1 NST = NST + 1

IF ((KCCCN4(55).EQ.L(39)).AND.(NI.EQ.55)) NST = NST - 2

IF ((KCCCN4(55).NE.BLNK).AND.(NI.EQ.55)) NST = NST + 2

KK = KCCCN4(NI)

215 CONTINUE

216 CONTINUE

NST = NST - 1

DO 217 NI = 8,55

NT1(NST) = KCCCN5(NI)

NST = NST + 1

IF ((KCCCN5(NI).EQ.BLNK).AND.(KK.EQ.BLNK)) GO TO 218

IF ((KCCCN5(53).EQ.L(39)).AND.(NI.EQ.53)) GO TO 218

IF ((KCCCN5(54).NE.BLNK).AND.(KCCCN5(55).EQ.BLNK).AND.(NI.EQ.55))

1 NST = NST + 1

IF ((KCCCN5(55).EQ.L(39)).AND.(NI.EQ.55)) NST = NST - 2

IF ((KCCCN5(55).NE.BLNK).AND.(NI.EQ.55)) NST = NST + 2

KK = KCCCN5(NI)

217 CONTINUE

218 CONTINUE

IF (NST.GE.NSP) GO TO 219

NST = NST + 1

NT1(NST) = BLNK

GO TO 218

C -----

219 CONTINUE

WRITE (NTAPE1) (NT1(J), J= 1,246)

C -----

C START ROUTINE FOR NTAPE2

DO 220 I = 1,200

220 NT2(I+37) = NT1(I+12)

C KEYWORD DETERMINATION

250 CONTINUE

DO 251 I=201,224

251 NTEST(I) = BLNK

DO 252 I= 1,200

252 NTEST(I) = NT2(I+37)

KO = 1

LCC = KO

C -----

257 CONTINUE

DO 253 I= 1,24

253 TMPKW(I) = BLNK

254 CONTINUE

DO 255 I = 1,24

TMPKW(I) = NTEST(KO )

IF (TMPKW(1).EQ.BLNK) GO TO 270

IF (TMPKW(I).EQ. BLNK) GO TO 258



## SOURCE STATEMENT

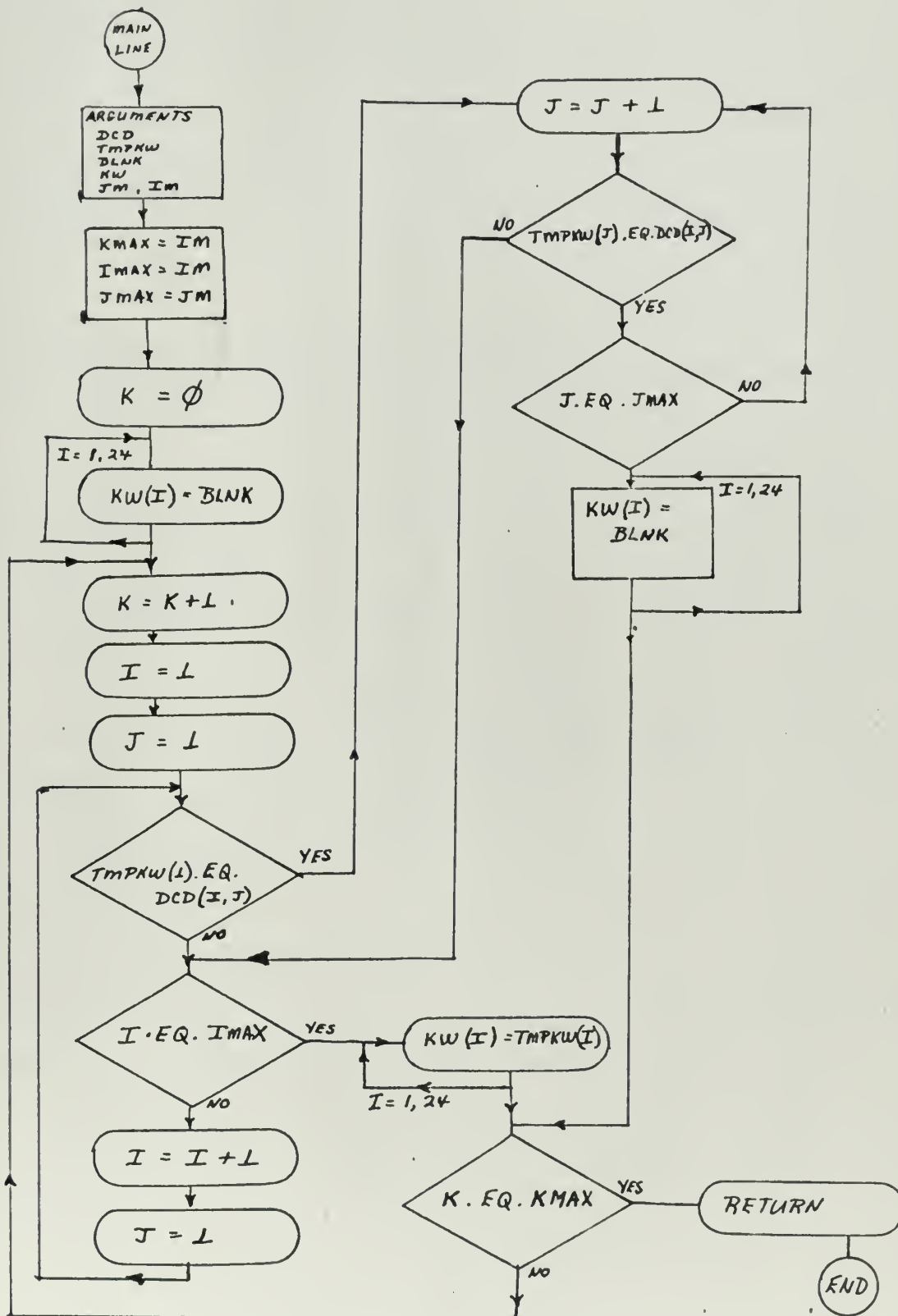
```

      KC = KC + 1
255 CONTINUE
C   NEXT 6 STMTS ADVANCE KC TO NEXT BLNK IN EVENT KEYWORD EXCEEDS 24 LTRS
      JJ = 0
259 JJ = JJ + 1
      JJJ = KC + JJ
      IF (NTEST(JJJ) .EQ. BLNK) GO TO 267
      GO TO 259
267 KC = KC + JJ
258 CONTINUE
C   HAVE POSSIBLE KEYWORD
      CALL KICK (DCC, TMPKW, BLNK, KW, JMX, IMX)
C   -----
      IF (KW(1).EQ.BLNK) GO TO 260
      DO 262 I = 1,24
262 NT2(I) = KW(I)
C   NEXT STMT STORES RELATIVE FIRST -LETTER LOCATION OF KEYWORD
      NT2(25) = LOC
263 CONTINUE
C   -----
261 CONTINUE
      WRITE (NTAPE2) (NT2(J), J=1,29)
      IF (KC .GE.200) GO TO 270
260 KC = KC + 1
      LOC = KC
      GO TO 257
C   -----
270 CONTINUE
C   START ROUTINE FOR NTAPE 3
C   WRITE (NTAPE3) (NT3(J), J=1,68)
      GO TO 1103
C   -----
1000 CONTINUE
      END FILE NTAPE1
      END FILE NTAPE2
      END FILE NTAPE3
      REWIND NTAPE1
      REWIND NTAPE2
      REWIND NTAPE3
      WRITE (6,281)
281 FORMAT (1H1,10X,17HMAINLINE COMPLETE, / , 11X,12HTAPES LOADED,
1 / , 11X, 13HTAPES REWOUND)
      CALL CLOCK (-3)
      CALL EXIT
      END

```



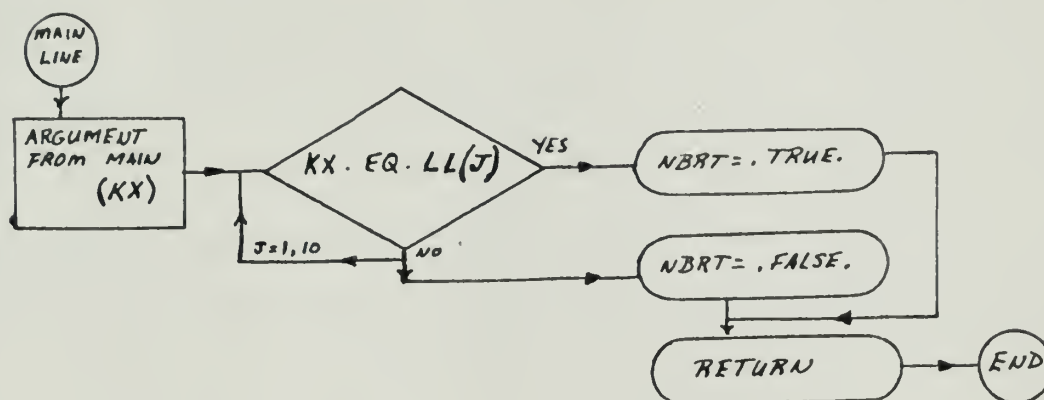
## FLOW CHART FOR SUBROUTINE KICK



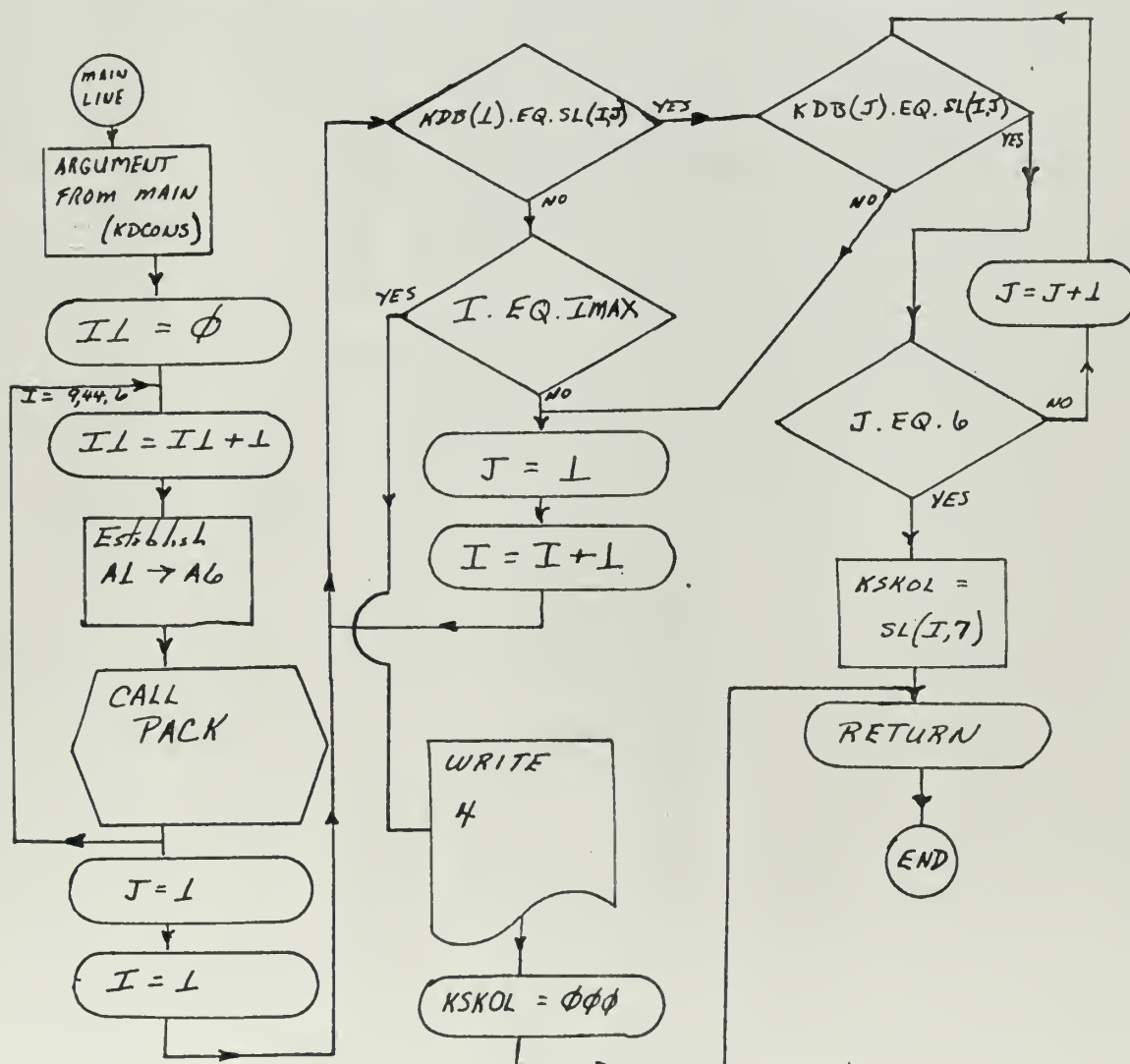




## FLOW CHART FOR NBRT FUNCTION SUBPROGRAM



## FLOW CHART FOR KSKOL FUNCTION SUBPROGRAM





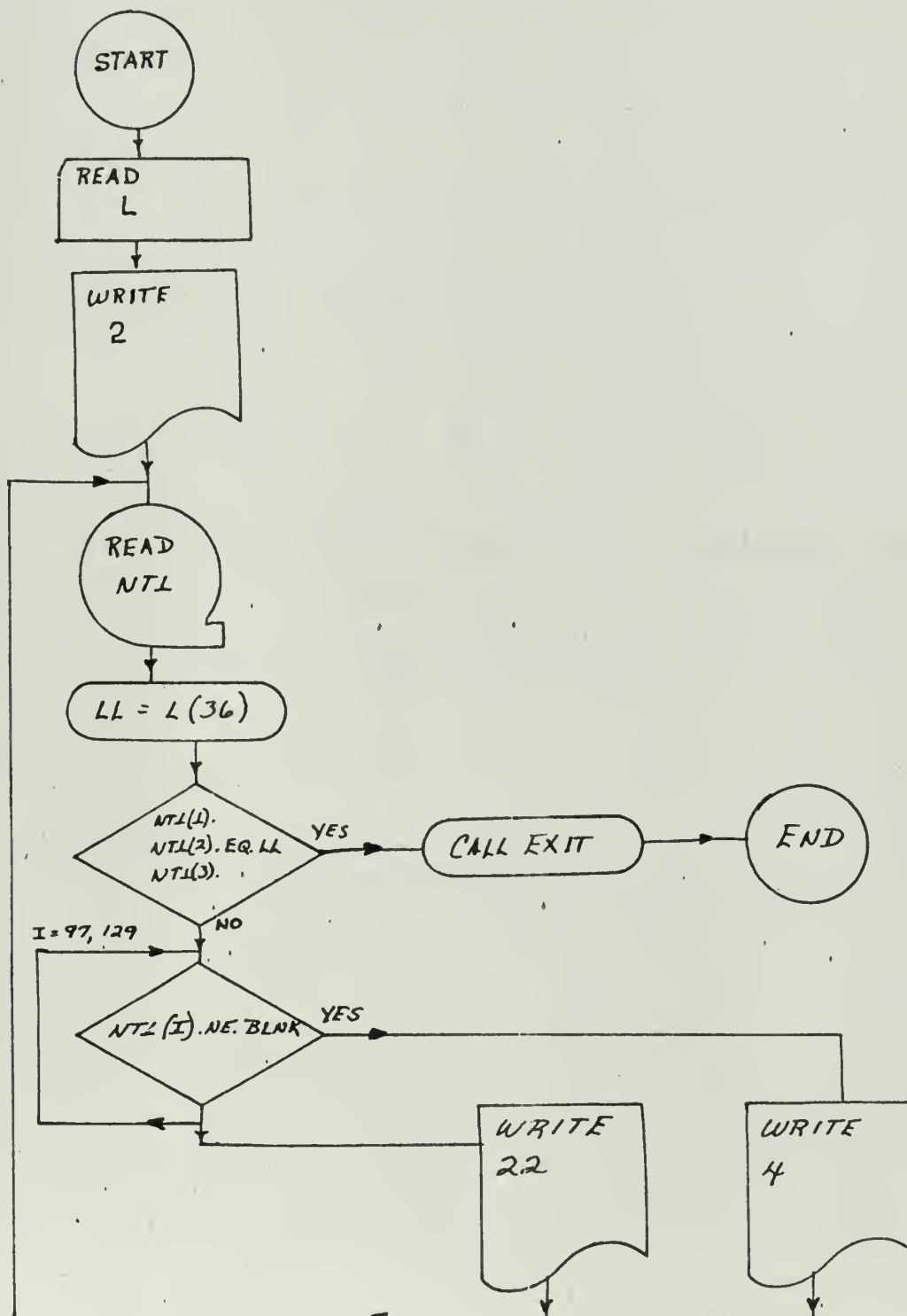


```
$IPFIC NBRT      NCDECK  
LOGICAL FUNCTION NBRT(KX)  
COMMON SL, IMAX, LL  
DIMENSION LL(47)  
DIMENSION SL(150,7)  
DO 1 J = 1,10  
IF (KX.EQ. LL(J))      GO TO 2  
1 CONTINUE  
NBRT = .FALSE.  
GO TO 3  
2 NBRT = .TRUE.  
3 CONTINUE  
RETURN  
END
```



```
$IBFTC KSKCL  NCDECK
      FUNCTION KSKCL (KDCCNS)
C      SUBPROGRAM TO RETURN SCHOOL CODE NUMBER
      INTEGER SL
      INTEGER A1,A2,A3,A4,A5,A6
      COMMON SL, IMAX, L
      DIMENSION SL(150,7) , KDCCNS(80) , KDB(6)
      DIMENSION L(47)
      I1 = 0
      DO 1 I2 = 9,44,6
      I1 = I1 + 1
      A1 = KDCCNS(I2)
      A2 = KDCCNS(I2 + 1)
      A3 = KDCCNS(I2 + 2)
      A4 = KDCCNS(I2 + 3)
      A5 = KDCCNS(I2 + 4)
      A6 = KDCCNS(I2 + 5)
      CALL PACK (KDB(I1) , A1,A2,A3,A4,A5,A6)
1 CONTINUE
      J = 1
      I = 1
2 CONTINUE
      IF (KDB(1).EQ.SL(I,J)) GO TO 6
      IF (I.EQ. IMAX) GO TO 3
5 CONTINUE
      J = 1
      I = I + 1
      GO TO 2
3 WRITE (6,4) (KDB(K) , K = 1,6)
4 FORMAT (10X, 34HSCHOOL NOT LISTED OR MISSPELLED - , ' 6A6 ')
      KSKOL = 000
      GO TO 50
6 CONTINUE
      IF (KDB(J) .EQ. SL(I,J)) GO TO 7
      GO TO 5
7 CONTINUE
      IF (J.EQ.6) GO TO 8
      J = J+1
      GO TO 6
8 CONTINUE
C HAVE FOUND SCHOOL IN DICTIONARY
      KSKOL = SL(I,7)
50 CONTINUE
      RETURN
      END
```



FLOW CHART FOR PROGRAM TO WRITE-OUT  
TAPE 1 — THE MASTER BIBLIOGRAPHY



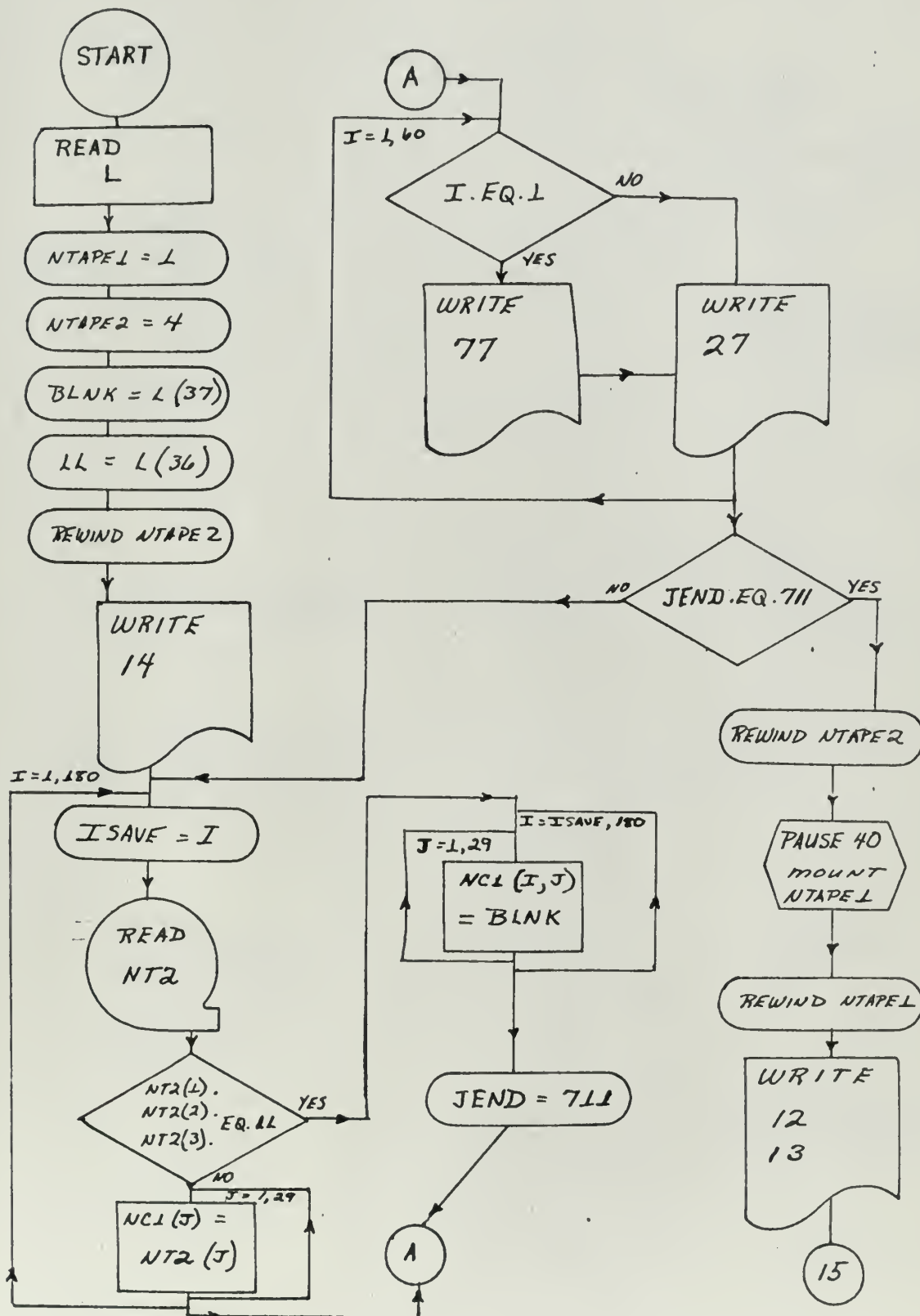
\$IBFIC RITE NCDECK

```
C PROGRAM TO WRITE NTAPE 1
  DIMENSION NT1(246), NT2(246), NT3(48), NWRITE(600)
  DIMENSION L(47)
  READ (5,1) L
1  FORMAT (47A1)
  INTEGER BLNK
  NTAPF1 = 4
  REWIND NTAPE1
  BLNK = L(37)
  WRITE (6,2)
2  FORMAT (1H1, 4CX, 35HMASTER RECCRD BIBLIOGRAPHIC LISTING, //)
  WRITE (6,3)
3  FORMAT (1X, 4HCCDE, 1X, 2HYR, 1X, 3HSCH, 20X, 5HTITLE, 61X, 6HAUTHOR, /)
5  READ (NTAPE1) (NT1(J), J= 1, 246)
  LL = L(36)
  IF ((NT1(1).EQ.LL) .AND.(NT1(2).EQ.LL) .AND.( NT1(3).EQ.LL))
1  GO TO 10
  DO 15 I = 97, 129
15  IF (NT1(I).NE.BLNK) GO TO 20
  GO TO 21
20  CONTINUE
  WRITE (6,4) ( NT1(I), I = 1, 246)
  4  FORMAT (1X, 8A1, A3, 120A1, / , 14X, 84A1, 1H*, 33A1)
  GO TO 5
21  CONTINUE
  WRITE (6,22) (NT1(I), I=1, 95), (NT1(J), J=214, 246)
22  FORMAT (1X, 8A1, A3, 86A1, 1H*, 33A1)
  GO TO 5
10  CONTINUE
40  CONTINUE
  REWIND NTAPE1
  PAUSE 10
  CALL EXIT
  END
```

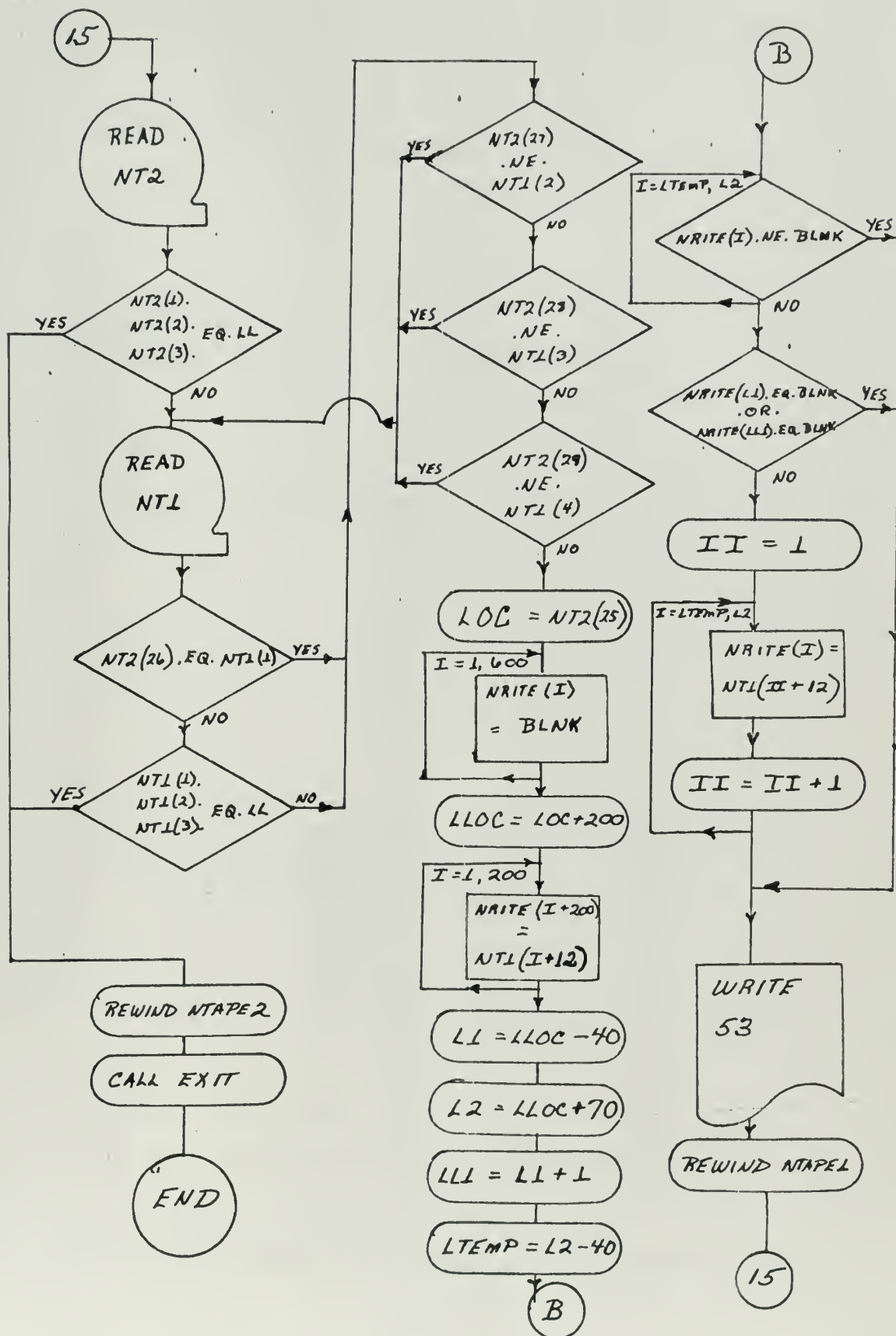




FLOW CHART FOR PROGRAM TO WRITE-OUT  
TAPE 2 AND PERMUTED-TITLE-INDEX









\$IRFIC RITE NCDECK

C PROGRAM TO WRITE NTAPE 2 AND KWIC

DIMENSION NT1(246), NT2( 29), NWRITE(600), NC1(180, 29)

DIMENSION L(47)

READ (5,1) L

1 FORMAT (47A1)

INTEGER BLNK

NTAPE1 = 1

NTAPE2 = 4

REWIND NTAPE2

BLNK = L(37)

LL = L(36)

C -----  
C ROUTINE FOR WRITING SORTED KEYWORD LIST

WRITE (6,14)

14 FORMAT (1H1,//////////, 20X, 32HWRITE OUT OF SORTED KEYWORD FILE)  
1E)

29 CONTINUE

CC 21 I=1,180

ISAVE = I

READ (NTAPE2) (NT2(J), J=1,29)

IF ((NT2(1).EQ.LL) .AND.(NT2(2).EQ.LL) .AND.(NT2(3).EQ.LL))GC TC23

CC 22 J=1,29

22 NC1(I,J) = NT2(J)

21 CONTINUE

GC TC 26

23 CC 25 I=ISAVE, 180

CC 25 J=1,29

25 NC1(I,J) = BLNK

JEND = 711

26 CONTINUE

CC 28 I= 1,60

IF (I.EQ.1) WRITE (6,77)

77 FORMAT (1H1)

WRITE (6,27)(NC1(I,J), J=1,29), (NC1(I+60,JJ),JJ=1,29),

1 (NC1(I+120,JJJ), JJJ=1,29 )

27 FORMAT (5X, 29A1, 6X, 29A1, 6X, 29A1)

28 CONTINUE

IF (JEND.EQ. 711) GO TO 30

GC TC 29

30 CONTINUE

REWIND NTAPE2

C -----  
PAUSE 40

REWIND NTAPE1

C ROUTINE FOR WRITING KWIC LIST

WRITE (6,12)

12 FORMAT (1H1,40X,20HPERMUTED TITLE INDEX,////////)

WRITE (6,13)

13 FORMAT (41X, 7HKEYWORD, 72X, 11HCODE-YR-SCH, /)

15 READ (NTAPE2) (NT2(J), J=1,29)

IF ((NT2(1).EQ.LL) .AND.(NT2(2).EQ.LL) .AND.(NT2(3).EQ.LL))

1 GO TO 20

C -----  
41 CONTINUE





## SOURCE STATEMENT

55

```

READ (NTAPE1) (NT1(J), J=1,246)
IF (NT2(26).EQ.NT1(1)) GO TO 42
IF ((NT1(1).EQ.LL).AND.(NT1(2).EQ.LL).AND.(NT1(3).EQ.LL))GO TO 20
GC TC41

```

```

42 CONTINUE
IF (NT2(27).NE.NT1(2)) GO TO 41
IF (NT2(28).NE.NT1(3)) GO TO 41
IF (NT2(29).NE.NT1(4)) GO TO 41
GC TC 45

```

```

45 CONTINUE

```

```

C -----
      LCC = NT2(25)
C   LCC IS RELATIVE LOCATION OF KW BETWEEN 1 AND 200

```

```

      CC 51  I = 1,600
51  WRITE (I) = BLNK
      LLCC = LCC +200
      CC 52  I = 1,200
52  WRITE (I+200) = NT1(I+12)
      L1 = LLCC-40
      L2 = LLCC +70

```

```

C   WRAP AROUND PORTION

```

```

      LL1 = L1 +1
      LTEMP = L2 - 40
      CC 80  I = LTEMP , L2
      IF (WRITE(I).NE. BLNK) GO TO 88
80  CONTINUE
      IF ((WRITE(L1).EQ.BLNK).OR.(WRITE(LL1).EQ.BLNK)) GO TO 88
      II = 1
      CC 81  I = LTEMP , L2
      WRITE(I) = NT1(II + 12)
      II = II+1

```

```

81 CONTINUE

```

```

88 CONTINUE

```

```

C -----
      WRITE (6,53) (WRITE(I), I=L1,L2) , (NT1(J), J=1,9 )

```

```

53  FORMAT ( 1X, 111A1, 8X, 8A1,A3 )

```

```

      REWIND NTAPE1

```

```

      GC TC 15

```

```

20 CONTINUE
      REWIND NTAPE2

```

```

      CALL EXIT

```

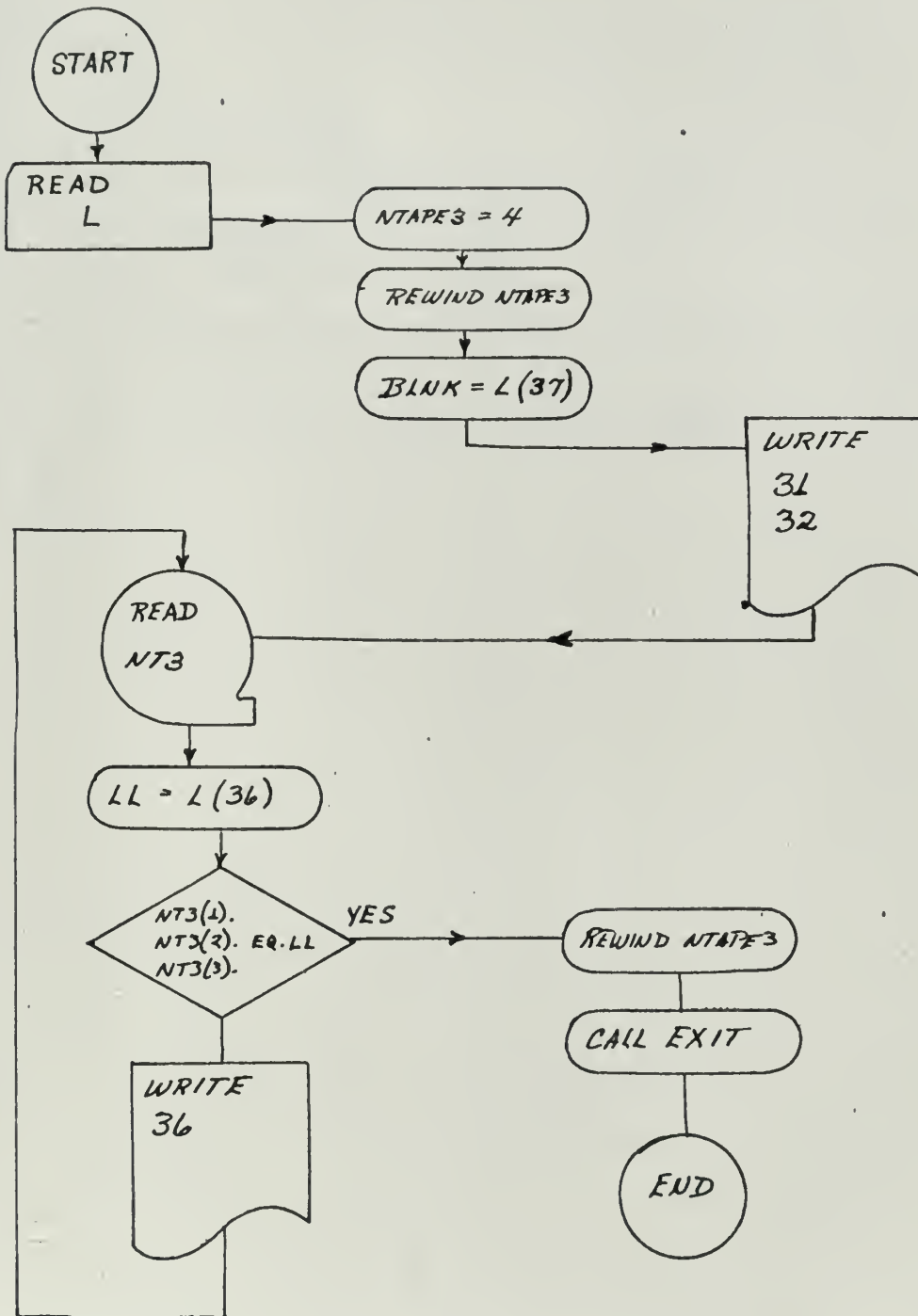
```

      END

```





FLOW CHART FOR PROGRAM TO WRITE - OUT  
TAPE 3 - THE AUTHOR LISTING



\$IBFTC RITE      NODECK

C    PROGRAM TO WRITE NTAPES 3

DIMENSION NT1(246), NT2(246),    NT3(68),    NRITE(600)

DIMENSION L(47)

READ (5,1)    L

1    FORMAT (47A1)

INTEGER BLNK

NTAPE3 = 4

REWIND NTAPE3

BLNK = L(37)

C    ROUTINE FOR AUTHOR LISTING

WRITE (6,31)

31    FORMAT (1H1, 40X, 14HAUTHOR LISTING, //)

WRITE (6,32)

32    FORMAT (10X, 6HAUTHOR,    40X, 4HCODE, / )

35    READ (NTAPE3)    (NT3(J), J= 1,68)

LL = L(36)

IF ((NT3(1).EQ.LL).AND.(NT3(2).EQ.LL).AND.(NT3(3).EQ.LL))

1    GO TO 40

WRITE (6,36)    (NT3(I),I=35,67), (NT3(J), J=21,29), (NT3(K),K=1,20)

36    FORMAT (10X, 33A1, 13X, 8A1,A3,    15X,    20A1)

GO TO 35

40    CONTINUE

REWIND NTAPE3

CALL EXIT

END



## APPENDIX A (continued)

C SUBROUTINE PACK

C -----

CALL PACK (NPAK, N1,N2,N3,N4,N5,N6 )

C -----

\$IBMAP	PAK	NODECK
	ENTRY	PACK
PACK	TRA	**
	SXA	GEORGE,4
	LAC	*-2,4
	PCS*	3,4,0
	SAC*	2,4,0
	PCS*	4,4,0
	SAC*	2,4,1
	PCS*	5,4,0
	SAC*	2,4,2
	PCS*	6,4,0
	SAC*	2,4,3
	PCS*	7,4,0
	SAC*	2,4,4
	PCS*	8,4,0
	SAC*	2,4,5
GEORGE	AXT	**,4
	TRA	PACK
	END	

C SUBROUTINE UNPACK

C -----

CALL UNPACK (NPAK, N1,N2,N3,N4,N5,N6 )

C -----

\$IBMAP	UNPAK	NODECK
	ENTRY	UNPACK
UNPACK	TRA	**
	SXA	GEORGE,4
	LAC	*-2,4
	PCS*	2,4,0
	SAC*	3,4,0
	PCS*	2,4,1
	SAC*	4,4,0
	PCS*	2,4,2
	SAC*	5,4,0
	PCS*	2,4,3
	SAC*	6,4,0
	PCS*	2,4,4
	SAC*	7,4,0
	PCS*	2,4,5
	SAC*	8,4,0
GEORGE	AXT	**,4
	TRA	UNPACK
	END	



## APPENDIX A (continued)

## CONTROL CARDS FOR SORT

```
$JOB          0694,DRURY
$PAUSE 20
$TIME         010
$IBSRT        NOTYPE
FILE,INPUT/2, REELS/1,  MODE/B,  BLOCKSIZE/030, UNLOAD
FILE,OUTPUT,  MODE/B,  BLOCKSIZE/030,  UNLOAD
RECORD, TYPE/F, LENGTH/030,  FIELD/(6,36B,36B,36B,36B,36B,36B,
1  36B,36B,36B,36B )
SORT, FILE/2, SEQ/C, ORDER/2, FIELD/(2,3,4,5,6,7,8,9,10,11)
SYSTEM,INPUTCHANNEL/S.SU04, OUTPUT CHANNEL/A, MERGE CHANNEL/A
OPTION , NODUMP, NOCKPT
END
```





## APPENDIX A (continued)

## DISCARD DICTIONARY

OF  
THE  
AND  
IN  
ON  
BY  
A  
WITH  
FOR  
FROM  
TO  
SOME  
AS  
AT  
AN  
Y.  
N.  
-  
.  
-  
STUDY  
DURING  
USING  
SURVEY  
EFFECT  
RELATION  
BETWEEN  
EXPERIMENT  
PHENOMENA  
PRODUCTS  
INTO  
TYPE  
RATE  
DROP  
ITS  
SYSTEMS  
ARISING  
DETAILED  
LOCAL  
REF  
APPLIED  
UNDER  
SIZE  
STUDIES  
PLANT



## Appendix B

### SAMPLE INPUT DATA

- A042 AMBROSE, TOMMY W. 1957  
LOCAL SHELL-SIDE HEAT TRANSFER COEFFICIENTS IN  
BAFFLED TUBULAR HEAT EXCHANGERS  
OREGON STATE COLLEGE
- A043 AMDUR, E. 1944  
THE DEHYDRATION OF VEGETABLES  
UNIVERSITY OF MINNESOTA
- A044 AMERIGO, FRANK TESI 1943  
KINETIC STUDIES OF THE ADSORPTION OF PHENOL BY  
ACTIVATED CARBON IN A PACKED TOWER  
UNIVERSITY OF PITTSBURGH
- A045 AMICK, E. H., JR. 1939  
PRESSURE-VOLUME-TEMPERATURE RELATIONSHIPS OF A  
METHANE-ISOPENTANE MIXTURE  
YALE UNIVERSITY
- A046 AMIR-YEGAHEH, A. 1962  
ON DETERMINATION OF VAPOR-LIQUID EQUILIBRIUM  
PHASE DISTRIBUTION RATIOS OF PETROLEUM OILS  
OKLAHOMA STATE UNIVERSITY
- A047 ANDERSEN, L. B. 1954  
ABSORPTION-OXIDATION PROCESSES IN DISPERSED MEDIA  
UNIVERSITY OF ILLINOIS
- A048 ANDERSON, D. G., JR. 1940  
VOID VOLUME AND FLOW RESISTANCE OF BEDS OF  
PARTICLES  
COLUMBIA UNIVERSITY
- A049 ANDERSON, DONALD KEITH 1959  
MUTUAL DIFFUSION IN ASSOCIATED BINARY LIQUID  
MIXTURES  
UNIVERSITY OF WASHINGTON
- A050 ANDERSON, F. A. 1947  
STUDY OF METHODS FOR PROCESSING TALL OIL  
LOUISIANA STATE UNIVERSITY
- A051 ANDERSON, J. E. 1953  
EVALUATION OF CATALYSTS FOR HYDROCARBON  
OXIDATION  
IOWA STATE UNIVERSITY
- A052 ANDERSON, J. E. 1955  
MASS TRANSFER IN A DISTILLATION COLUMN  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
- A053 ANDERSON, J. W. 1950  
PREPARATION OF NITROGENOUS FERTILIZERS BY  
TREATMENT OF CELLULOSIC MATERIALS WITH AMMONIA  
IOWA STATE UNIVERSITY





073	TEAS COLLEGE OF ARTS AND INDUSTRY
074	TEXAS TECHNOLOGICAL COLLEGE
075	TRI-STATE COLLEGE
076	UCLTS UNIVERSITY
077	TULANE UNIVERSITY
078	UNIVERSITY OF ALABAMA
079	UNIVERSITY OF ALBERTA
080	UNIVERSITY OF ARIZONA
081	UNIVERSITY OF ARKANSAS
082	UNIVERSITY OF BRITISH COLUMBIA
083	UNIVERSITY OF CALIFORNIA AT BERKELEY
084	UNIVERSITY OF CALIFORNIA AT LOS ANGELES
085	UNIVERSITY OF CINCINNATI
086	UNIVERSITY OF COLORADO
087	UNIVERSITY OF CONNECTICUT
088	UNIVERSITY OF DAYTON
089	UNIVERSITY OF DELAWARE
090	UNIVERSITY OF DENVER
091	UNIVERSITY OF DETROIT
092	UNIVERSITY OF FLORIDA
093	UNIVERSITY OF HOUSTON
094	UNIVERSITY OF IDAHO
095	UNIVERSITY OF ILLINOIS
096	UNIVERSITY OF KANSAS
097	UNIVERSITY OF KENTUCKY
098	UNIVERSITY OF LOUISVILLE SPEED SCIENTIFIC SCHOOL
099	UNIVERSITY OF MAINE
100	UNIVERSITY OF MARYLAND
101	UNIVERSITY OF MASSACHUSETTS
102	UNIVERSITY OF MICHIGAN
103	UNIVERSITY OF MINNESOTA
104	UNIVERSITY OF MISSISSIPPI
105	UNIVERSITY OF MISSOURI
106	UNIVERSITY OF MONTREAL
107	UNIVERSITY OF NEBRASKA
108	UNIVERSITY OF NEW HAMPSHIRE
109	UNIVERSITY OF NEW MEXICO
110	UNIVERSITY OF NORTH DAKOTA
111	UNIVERSITY OF NOTRE DAME
112	UNIVERSITY OF OKLAHOMA
113	UNIVERSITY OF OMAHA
114	UNIVERSITY OF PENNSYLVANIA
115	UNIVERSITY OF PITTSBURGH
116	UNIVERSITY OF PUERTO RICO
117	UNIVERSITY OF RHODE ISLAND
118	UNIVERSITY OF ROCHESTER
119	UNIVERSITY OF SASKATCHEWAN
120	UNIVERSITY OF SOUTHWESTERN LOUISIANA
121	UNIVERSITY OF SOUTH CAROLINA
122	UNIVERSITY OF SOUTHERN CALIFORNIA
123	UNIVERSITY OF TENNESSEE
124	UNIVERSITY OF TEXAS
125	UNIVERSITY OF TOLEDO
126	UNIVERSITY OF TORONTO
127	UNIVERSITY OF UTAH
128	UNIVERSITY OF UTAH
129	UNIVERSITY OF VIRGINIA
130	UNIVERSITY OF WASHINGTON
131	UNIVERSITY OF WISCONSIN
132	VANDERBILT UNIVERSITY
133	VILLANOVA UNIVERSITY
134	VIRGINIA POLYTECHNIC INSTITUTE
135	WASHINGTON STATE UNIVERSITY
136	WASHINGTON STATE UNIVERSITY
137	WAYNE STATE UNIVERSITY
138	WEST VIRGINIA INSTITUTE OF TECHNOLOGY
139	WEST VIRGINIA UNIVERSITY
140	WORCESTER POLYTECHNIC INSTITUTE
141	YALE UNIVERSITY
142	UNIVERSITY OF LOUISVILLE
143	THE INSTITUTE OF PAPER CHEMISTRY
144	UNIVERSITY OF IOWA

[illegible]

CODE	YR	SCN	TITLE	AUTHOR
A001	37	102	EFFECT OF INITIAL VELOCITY DISTRIBUTION ON HEAT TRANSFER IN SPEEDY TUBES	ANBARUCHI, PETER HERMAN
A002	58	039	THE DEVELOPMENT OF CALCIUM CARBONATE-IRON OXIDE PROTECTIVE COATINGS ON IRON	ANDERSON, P. U.
A003	30	115	LIQUID PHASE CRACKING OF PETROLEUM	ANDERSON, RAYMOND LESTER
A004	37	058	THE DENSITY OF MIXTURES OF CARBON DIOXIDE AND ARGON AT 50 DEGREES AND 50 TO 1000 ATMOSPHERES	ANDERSON, W. H.
A005	37	054	THERMODYNAMIC CORRELATION AND PREDICTION OF VAPOR-LIQUID EQUILIBRIA FOR ETHYLENE ACETATE-CYCLOPENTANE MIXTURES AND ACETATE-CYCLOPENTANE MIXTURES	ANDERSON, J. A.
A007	62	060	THE MECHANISM OF CRYSTALLINE POLYMER MATERIALS IN THE FALLING RATE PERIOD TREATMENT OF MIXTURES WITH AN INDEFINITE NUMBER OF COMPONENTS	ANDERSON, EDWARD F.
A008	28	037	THE REACTION OF LINE AND WATER	ANDERSON, F. W.
A009	61	000	VAPOR PHASE NITRATION OF OILS IN A FUSED SALT REACTOR	ANDERSON, FRANK SLATES JR.
A010	62	130	A STUDY OF THE CRITICAL HEAT FLUX IN AN ACCELERATED POOL BOILING SYSTEM	ANDERSON, J. P.
A011	50	055	CHEMICAL REACTION RATES IN HIGH PRESSURE	ANDERSON, H. H.
A012	42	026	THE VAPOR PHASE OXIDATION OF N-ETHYL MERCAPTAN OVER VANADIUM PENTOXIDE CATALYSTS	ANDERSON, GILBERT IRVING
A013	48	037	HEAT TRANSFER AT HIGH RATES TO WATER BOILING OUTSIDE CYLINDERS	ANDERSON, J. N.
A014	61	048	HEAT TRANSFER IN CONDENSATED POLYMER MEDIA WITH FLOWING FLUIDS	ANDERSON, RACHANANDH
A015	61	045	THE FLUIDIZATION OF UNIFORM SMOOTH SPHERES IN LIQUID MEDIA	ANDERSON, IRWIN LEROY
A016	50	131	ATONIZATION OF WATER WITH SPINNING DISKS	ANDERSON, CHARLES RICHARD
A017	59	072	THE FORMATION OF COAGULATES IN SATURATED MIXTURE OF KNO <sub>3</sub> AND STEAM	ANDERSON, LEONARD B.
A018	57	031	MELTING TIME FUNCTIONS OF SHORT TUBULAR VESSELS	ANDERSON, R. J.
A019	51	055	SOLVENT EXTRACTION OF VEGETABLE OILS	ANDERSON, JACOBSON
A020	48	130	THE CELLULOSE AND LIGNIN OF ADONIS BAMBOO WOOD	ANDERSON, AGGARWALA, JACOBSON PHANUURA
A021	49	037	IGNITION OF HIGH VELOCITY GAS STREAKS BY BEAMS OF HEATED ROSES	ANDERSON, G. A.
A022	59	102	A LIGHT SCATTERING STUDY OF UNFRACTIONATED POLYVINYL ACETATE	ANDERSON, RICHARD A.
A023	57	072	FREE SCISSORING OF FIBERS IN FLUIDS	ANDERSON, ANDERSON R.
A024	51	102	A KINETIC STUDY OF THE SYNTHESIS OF METHANE	ANDERSON, WILLIAM WALTER
A025	41	037	HEAT TRANSFER IN SUBMERGED TUBE EVAPORATION	ANDERSON, G. A.
A026	62	048	EXPERIMENTAL STUDY OF LAMINAR HEAT TRANSFER WITH PRESCRIBED WALL HEAT FLUX	ANDERSON, RICHARD C.
A027	50	050	LIQUID-VAPOR EQUILIBRIUM COMPOSITION RELATIONSHIPS IN THE ETHANE-CYCLOHEXANE SYSTEM	ANDERSON, RUALAT CYER
A028	53	139	PRESSURE DROP IN FLOW OF DENSE COAL-AIR MIXTURES	ANDERSON, CHARLES W.
A029	50	102	THERMODYNAMIC PROPERTIES OF FROST 13	ANDERSON, LYLE FREDERIC
A030	15	014	THE TIME FACTOR IN MAKING OIL GAS	ANDERSON, C. H.
A031	52	058	A NEW METHOD FOR STUDYING CHANNELING IN A PACKED GAS ADSORPTION TOWER	ANDERSON, L. C. JR.
A032	48	058	A STUDY OF THE VAPOR-PHASE NITRATION OF PARAFFINS	ANDERSON, L. C.
A033	60	039	MASS TRANSFER RATES IN PACKED BEDS	ANDERSON, L. C.
A034	32	016	LUBRICATING PROPERTIES OF CRACKED FRESH PETROLEUM OILS	ANDERSON, EMMETT EMMETT
A035	48	102	THE THROUGH-DRYING OF POROUS MEDIA	ANDERSON, HAROLD OSMAN
A036	32	014	THE ENGINEERING DEVELOPMENT OF PHOTOVOLTAIC CELLS	ANDERSON, JOSEPH
A037	54	103	THE TRANSIENT BEHAVIOR OF SINGLE-PHASE, NATURAL-CIRCULATION LOOP SYSTEMS	ANDERSON, U. K.
A038	54	045	ADSORPTION EQUILIBRIA OF OXYGEN AND NITROGEN ON VARIOUS ADSORBENTS	ANDERSON, C. D.
A039	34	131	SOURCE REQUIREMENTS FOR THE COMBUSTION OF CARBONIZED WATER GAS IN LUMINOUS FLAMES	ANDERSON, M.
A040	24	154	ADSORPTION OF VAPORS BY METALLIC OXIDES. ITS BEARING ON THE CATALYSIS OF ESTER FORMATION	ANDERSON, RIGGS JAMES
A041	62	068	THE RATE OF PYROLYSIS OF NITROGEN DIOXIDE IN A SHOCK TUBE	ANDERSON, A. M.
A042	57	053	LOCAL SHELL-SIDE HEAT TRANSFER COEFFICIENTS IN DAFLOD TUBULAR HEAT EXCHANGERS	ANDERSON, FREDERICK N.
A043	44	103	THE OXIDATION OF VEGETABLES	ANDERSON, L. H.
A044	43	115	KINETIC STUDIES OF THE ADSORPTION OF PHENOL BY ACTIVATED CARBON IN A PACKED TOWER	ANDERSON, FRANK LESI
A045	39	141	PRESSURE-VOLUME-TEMPERATURE RELATIONSHIPS OF A METHANE-ISOPENTANE MIXTURE	ANDERSON, C. H. JR.
A046	62	052	ON DETERMINATION OF VAPOR-LIQUID EQUILIBRIUM PHASE DISTRIBUTION RATIOS OF PETROLEUM OILS	ANDERSON, A.
A047	54	095	ABSORPTION-OXIDATION PROCESSES IN DISPERSED MEDIA	ANDERSON, L. H.
A048	40	014	VOID VOLUME AND FLOW RESISTANCE OF BEDS OF PARTICLES	ANDERSON, O. G. JR.
A049	59	130	MUTUAL DIFFUSION IN ASSOCIATED BINARY LIQUID MIXTURES	ANDERSON, DONALD ALVIN
A050	47	033	STUDY OF METHODS FOR PROCESSING TELLURITE	ANDERSON, F. A.
A051	53	025	EVALUATION OF CATALYSTS FOR HYDROCARBON OXIDATION	ANDERSON, J. E.
A052	55	037	MASS TRANSFER IN A DISTILLATION COLUMN	ANDERSON, J. E.
A053	50	025	PREPARATION OF NITROGENOUS FERTILIZERS BY TREATMENT OF CELLULOSIC MATERIALS WITH AMMONIA	ANDERSON, J. E.
A054	49	131	INTERNAL BURNING VELOCITIES AND FLAME TEMPERATURES OF LAMINAR PROPANE AIR FLAMES FORM ICE TRACKS	ANDERSON, J. E.
A055	41	131	KINETICS OF THE HYDROGENATION OF 1,5-OCTADIENE	ANDERSON, J. E.
A056	62	130	THERMODYNAMIC STUDIES IN THE SYSTEM CELLULOSE-WATER. STUDIES IN THE SYSTEM CELLULOSE-WATER	ANDERSON, NELS KENNETH
A057	53	060	PRESSURE DROPS ACROSS SUBMERGED ORIFICES WITH PSEUDO-PLASTIC NON-NEWTONIAN FLUIDS	ANDERSON, NELS KENNETH
A058	61	083	PHYSICOCHEMICAL PROPERTIES OF POLAR-NONPOLAR LIQUID MIXTURES	ANDERSON, R. A.
A059	38	037	INTERFACIAL TENSION BY PENDANT DROPS	ANDERSON, RALPH
A060	42	112	VAPOR-LIQUID EQUILIBRIA IN A BINARY-FLUID-SOLVENT	ANDERSON, J. H.
A061	35	025	RECOVERY OF URANIUM FROM SUPERPHOSPHATE	ANDERSON, DAN L.
A062	52	001	THE FORMATION OF CHEMICAL FILM DEPOSITS ON ALUMINUM SURFACES	ANDERSON, A. W.
A063	60	131	HYDROGENATION WITH CALCIUM CYCLOPENTADIENES	ANDERSON, VINCENTI
A064	52	026	AN EQUATION FOR THE ADSORPTION WAVE	ANDERSON, DOROTHY E.
A065	61	102	THE ELECTROLYSIS OF SOME LIQUID ALKYL HALIDES	ANDERSON, WALTER JOHN
A066	62	057	CHARACTERIZATION OF MULTIPLE VARIABLE PROCESS SYSTEMS FROM RANDOM INPUTS	ANDERSON, JOHN C.
A067	53	144	HEAT TRANSFER IN A FLUIDIZED SOLIDS SYSTEM	ANDERSON, RICHARD N.
A068	39	037	HYDRATION OF ETHYLENE BY BEANS OF MINERAL ACIDS AT HIGH PRESSURES	ANDERSON, J. R.
A069	53	089	REACTION, FLOW, AND RADIATION IN ENCLOSED, PREDISTABILIZED FLAMES AT LOW PRESSURE AND HIGH VELOCITIES	ANDERSON, R. A.
A070	32	058	REACTION KINETICS AND HEAT TRANSFER IN PACKED BED	ANDERSON, W. B.













thesD782

A FORTRAN program for permuted title ind



3 2768 000 98563 4

DUDLEY KNOX LIBRARY